Covenant Journal of Informatics & Communication Technology

Vol.3 No. 2, December 2015

A Bi-annual Publication of the Departments of Computer Information Science, and Electrical & Information Engineering
Covenant University, Canaan Land,
Km 10, Idiroko Road, Ota, Ogun State, Nigeria.

Editor-in-Chief: **Prof. Sanjay Misra**
sanjay.misra@covenantuniversity.edu.ng

Managing Editor: **Edwin O. Agbaike**
edwin.agbaike@covenantuniversity.edu.ng

Website: http//Journal.covenantuniversity.edu.ng

It is a condition of publication in this journal that manuscripts have not been published or submitted for publication and will not be submitted or published elsewhere.

Upon the acceptance of articles to be published in this journal,the author(s) are required to transfer copyright of the article to the publisher.

ISSN - Print: 2354 – 3566
- Electronics: 2354 – 3507

Published by Covenant University Journals,
Covenant University, Canaanland, Km 10, Idiroko Road,
P.M.B. 1023, Ota, Ogun State, Nigeria.

Printed by Covenant University Press.

Articles

Equalization of DS-UWB Systems using Genetic Algorithm with Adaptive Parameters

Nazmat Surajudeen-Bakinde[1]

&

Xu Zhu[2]

[1]Department of Electrical and Electronics Engineering, University of Ilorin,
Ilorin, Nigeria deenmat1211@gmail.com,

[2]Department of Electrical Engineering and Electronics, University of Liverpool,
United Kingdom
xuzhu@liverpool.ac.uk

Abstract— We use adaptive generation along with some other parameters to investigate their effects on the performance of Genetic Algorithm (GA) in comparison to a previous work, where the output of a RAKE receiver is utilized as the input to a GA so as to reduce the inter-symbol interference (ISI) due to the frequency selectivity of UWB channels because of the very high rate of transmission. The effects of two different scaling methods and two mutation types, on the performance of a GA when used with a receiver for the equalization of the channel of a direct sequence ultra wideband (DS-UWB) wireless communications system are presented. The results show that fitness scaling has effects on GA based optimization while mutation prevents local convergence.

Keywords—Genetic algorithm, stall generation, function tolerance, fitness limit, adaptive parameters.

1. **Introduction**

Ultra wideband (UWB) communication has been defined as one of the most promising technologies, where the very large bandwidth of 3.1−10.6 GHz allows UWB to be applicable for communication systems that are innovative and at the same time transmitting at a very fast rate and an efficient manner (Perarasi & Ravichandran, 2014). In (Somayazulu et al., 2002), the data rate and the coverage distance of UWB as a promising technology was given as 110 Mbps to 480 Mbps at distance of 2 m to 10 m respectively. The low power spectral density of UWB and its benefits was also explained by (Nassar et al, 2003.).

The application of RAKE receivers to UWB was done earlier in 2005 by (Sato & Ohtsuki, 2005), where it was shown that when maximum ratio combining when used with RAKE receiver has low computational complexity even though it was a perfect channel estimation that was considered. In another related research work by (Siriwongpairat & Liu, 2008), narrowband interference (NBI), ISI and multipath fading are challenges for a channel that the fading is frequency-selective and so all these severely degrade the performance of the UWB system considered. It was concluded in the work that multipath diversity can be exploited by the constructive combination the monocycles that was obtained from the paths that are resolvable.

In previous work by these authors (Surajudeen-Bakinde et al, 2009), the lower computational complexity of genetic algorithm as optimization technique used for UWB, was shown to be much lower than maximum likelihood detection approach.

II. Related Work

Optimization techniques as applied to UWB was taken to different level by (Montaser et al., 2013), where Modified Particle Swarm Optimization (MPSO) and Bacterial Swarm Optimization (BSO) in addition to Central Force Optimization (CFO) was as recent optimization technique to optimize a notched-UWB E-shaped patch antenna. The return loss, antenna gain and radiation patterns which are the antenna parameters were also discussed.

For the enhancement of a multi-objective genetic algorithm, a machine learning technique was applied by (Martins et al, 2012), whereby a microstrip antenna used in ultra-wideband (UWB) wireless devices, was analyzed so as to know the estimates of the fitness function behaviours, from a set of experiments made in a laboratory. A novel genetic algorithm (GA) that is based on complementary error function mutation (CEFM), was used as a differential multiuser detection (MUD) method for ultra-wideband (UWB) systems by (Qi et al., 2011). In another research, a multiuser detection (MUD) method using a novel genetic algorithm (GA) based on complementary error function mutation (CEFM) and a differential algorithm (DA) for ultra-wideband (UWB) systems was proposed by (Kong et al, 2011). Six existing forms of fitness scaling in genetic algorithms were presented, as the first systematic means of evaluating the effects of the fitness scaling functions was compared to a new method that is called transform ranking. The number of generations used was fixed for the stochastic universal sampling which was applied individually. (Hopgood & Mierzejewska, 2009).

The statistical analysis of the minimum mean square error (MMSE) was done for the investigation of the problem encountered in the finger selection process for a UWB Selective Rake receiver. An iterative scheme based on GA was proposed because the optimal solution is NP hard (Gezici et al., 2005). B-spline basis approximation was used as a flexible method of designing UWB pulse waveform by some researchers, (Wang et al., 2008). The power spectral mask of the Federal Communications Commission (FCC) for indoor UWB systems must be met in addition to ensuring that in the design of UWB, the orthogonality is also preserved at the correlation receiver (Wang et al. 2008).

Hill and the other researchers in (Hill et al., 2004), made a comparison between the effectiveness of using fitness scaling in a GA and using an inversion operator in a GA. In (Sadjadi, 2004), four scaling methods were compared based on their handling of a simple set of optical processing data. Also in (Kreinovich et al., 1993), problem of choosing a scaling function as a mathematical optimization was formulated.

In all the above researches in GA as optimization techniques and its application to UWB, no work has been done in investigating the performance of applying adaptive parameters to channel equalization of DS-UWB communication systems using GA. In this work, we use adaptive parameters to know their effects on the performance of GA as an optimization technique in comparison to a previous work (Surajudeen-Bakinde et al., 2009). A plot of the bit-error-rate (BER) versus the size of the generation used is obtained to show the rate at optimization is converging. The effects of two different scaling methods and mutation types on the performance of the channel equalization of the DS-UWB systems using a GA as an optimization technique was also investigated. This work is an extention of work done in (Surajudeen-Bakinde et al., 2009).

This paper is organized such that we reviewed work done in the same field of research in Section II. Section III is the system model. The equalization of DS-UWB system using GA is given in Section IV. The results obtained from the simulation is given is in Section V and finally, the paper is concluded in Section VI.

III. System Model

A. Signal Transmitted

The ternary orthogonal code sequence which is the transmit pulse vTR (t), is generated in accordance to the expression given thus:

$$v_{TR}(t) = \sum_{i=0}^{N_c-1} b_i g(t - iT_c) \qquad (1)$$

where the length of the spreading code is N_c, *the* i_{th} component of the spreading code is b_i, the chip width is T_c and represents the transmitted monocycle waveform that is already normalized to have unit energy is $g(t)$

The expression that follows is for the DS-UWB signal:

$$x(t) = \sqrt{E_c} \sum_{k=-\infty}^{\infty} d_k v_{TR}(t - kT_f) \qquad (2)$$

where the energy for each transmitted pulse is E_c , the d_k ∈ {±1} is the k_{th} transmit symbol, T_f is the frame time and each of the frame considered is further divided into N_c which are equally spaced chips that give rise to $T_f = N_c T_c$.

B. Channel

According to (Foerster, 2003), the UWB channel model which was derived from the Saleh-Valenzuela model and has undergone some modifications is employed the simulation done in this work. Instead of using a Rayleigh distribution, the model used a log-normal distribution for the multipath gain magnitude and this was because it is found to have a better fit the measurement data. Also, an assumption of independent fading is taken for the individual cluster and equally for each ray within the cluster. In a simpler form, the channel impulse response of the Saleh-Valenzuela model is given thus:

$$h(t) = \sum_{l=1}^{L_{tot}} h_l \delta(t - \tau_l) \quad (3)$$

where the total number of paths for this work is L_{tot}, with a delay τ_l (= lT_c) for the l_{th} component, where the l_{th} path gain is h_l (Foerster, 2003).

C. Signal Received

The signal received as a result of the convolution of the signal transmitted which is given in (2), with the defined channel impulse responses in discrete time, given in (3) and then the addition of the noise component, which is the additive white Gaussian noise (AWGN), is given in the equation that follows thus:

$$r(t) = x(t) * h(t) + n(t)$$

$$= \sqrt{E_c} \sum_{k=-\infty}^{\infty} d_k v_{TR} \sum_{l=1}^{L_{tot}} h_l \left(t - kT_f - \tau_l\right) + n(t)$$

(4)

where $n(t)$ is the AWGN, having a variance of σ^2 and zero mean.

IV. Channel Equalization of DS - UWB Communication System Using GA

A. RAKE Receiver

The importance of a good initial value in GA based algorithms was put into consideration in this work. When GA only, without proper initialization was applied to DS-UWB systems, the performance is worse than RAKE receiver only. This is because of the frequency selective nature of UWB channels. For this reason, the RAKE output soft estimates was used as the initial population for the channel equalization so as to have an improved performance of our system.

(Siriwongpairat & Liu, 2008) defined a typical RAKE receiver as one that has many correlators which is then followed by a linear combiner. An assumption of having a channel without requiring estimation at perfect chip synchronization is also assumed to be between the transmitter and the receiver. The output of the correlator which is expressed in vector is expressed in the next expression.

$$y_k = \sqrt{E_s} d_k h + i_k + n_k \qquad (5)$$

where $y_k = \left[y_k^{f1}, \ldots y_k^{fL}\right]^T$, $E_s = N_c E_c$, which is the energy per symbol,

$h = \left[h_{f1},...h_{fL}\right]^T, i_k = \left[i_k^{f1},...i_k^{fL}\right]^T$ with i_k^{fL} denoting ISI of the k_{th} symbol for the l_{th} correlator and $n_k = \left[n_k^{f1},...n_k^{fL}\right]^T$ with n_k^{fL} is the noise component of the k_{th} symbol for the correlator l_{th}. The number of RAKE fingers used is symbolized with L

B. Genetic Algorithm

According to (Man et al., 1999), the definition of GA was given as a technique that works on the Darwinian principle of natural selection which is referred to as "survival of the fittest". The researchers went on to explain that the degree of goodness of the chromosome being considered for the problem, which in any case would be highly related with its objective value is shown by the fitness value. A better solution is more likely to emerge from a fitter chromosome, which invariably yield good quality offspring, throughout a genetic evolution. The number of possible solutions, searched during the optimization process using the GA, are specified, before refining is done using the GA operators. Minimization of the fitness function is done by the GA in terms of the distance measure criteria according to the cost function given thus:

$$J = \left|\gamma^T e\right| \; J = \left|\gamma^T e\right| \qquad (6)$$

where

$e = \left[e_1...,e_M\right], e_k = y_k - \sum_{l=1}^{L_{tot}} h_1 d(k-1)$ and $k = 1 - M$. $\gamma = \left[\gamma_1,...\gamma_L\right]^T$ is the weights of the finger for the RAKE receiver which is being estimated from the channel taps that is given as, $\gamma_l = \left\lfloor h_{fl} \right\rfloor$. The other terms have already being defined in the RAKE-GA initialization section.

V. Fitness Scaling and Adaptive Parameters for GA Based Equalization

A. Types of Fitness Scaling

Fitness scaling converts the raw fitness scores that are

returned by the fitness function to values in a range that is suitable for the selection function. The selection function assigns a higher probability of selection to individuals with higher scaled values. The range of the scaled values affects the performance of the GA. Two scaling options compared in this paper are rank and proportional methods.

- Expectation is proportional to the scores of the raw fitness in a GA when **proportional fitness scaling is implemented**. The advantage to this is beneficial when you have good range for raw scores. When the scaled values do not vary too much as the individuals with the highest scaled values reproduce too rapidly, a good and suitable range is thus obtained so as to avoid it taking over the population gene pool to quickly, and preventing the genetic algorithm from searching other areas of the solution space. Also the range should vary a widely so as let the individuals have almost the same chance of reproduction and the search will equally progress quite slowly (MATLAB, 2007).

• The raw score using the **rank fitness scaling**, is scaled when it is based on the rank of each member and not the score. The position in the sorted scores is rank of an individual and the fittest individual is ranked 1, the fitter individual is has a rank of 2, and so on. A scaled value is assigned such that, the scaled value of an individual with rank n is proportional to $\frac{1}{\sqrt{n}}$. The effect of the spread of the raw scores is thus removed (MATLAB, 2007).

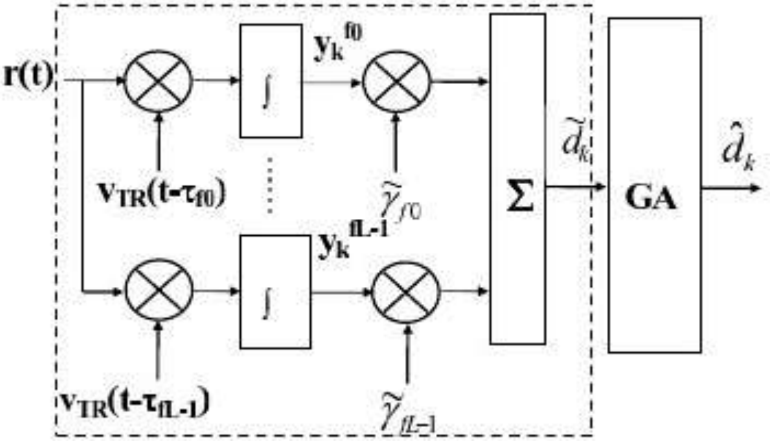


Figure 1. Block Diagram of RAKE-GA for DS-UWB

B. Operators in RAKE-GA for DS-UWB System

The block diagram of the RAKE receiver, as being the input to the RAKE in combination with the GA for the DS-UWB system is presented in Fig. 1. The three main GA operators implemented in this work are as described thus:

• **Stochastic selection** is the method used to choose parents, based on their scaled values from the fitness scaling function, for the next generation. A line is now laid out, whereby each parent correspond to a section of the line of length proportional to its expectation. Elites that are guaranteed to survive to the next generation are then chosen.

• Two parents come together to produce a child for the next generation when **scattered crossover** is implemented in a GA operation. The genes are selected by the created binary vector, whereby the genes where the vector is a 1 comes from the first parent, and the genes where the vector is a 0 comes from the second parent. The combination subsequently takes place for a child to be formed.

• **Gaussian mutation and uniform mutation** are two different mutation types employed in this work. Convergence to a local minimum point is prevented by the mutation types. A random number, which has a mean of zero is added from a

Gaussian distribution to each of the vector entry of an individual. The scale and shrink parameters are used to control the variance of the distribution. The variance at the first generation is determined by the scale parameter, while shrinking of the variance as generations go is controlled by the shrink parameter. Uniform mutation on the other hand is a two-step process, firstly the algorithm selects a fraction of the vector entries of an individual for mutation, where each entry has the same probability as the mutation rate of being mutated. Secondly, the algorithm replaces each selected entry by a random number selected uniformly from the range for that entry (MATLAB, 2007).

C. Adaptive Parameters in RAKE-GA for DS-UWB

The algorithm is terminated by setting the **stopping criteria** to terminate. The maximum generations for the algorithm, which was fixed in our previous work (Surajudeen-Bakinde et al., 2009) is termed the **Generations.** Three more stopping criteria, namely, fitness limit, stall generations and function tolerance are implemented in this work. The algorithm is stopped, by the fitness limit. This occurs for the best point in the current population, when the value of the fitness function is less than or equal to the specified fitness limit value. When the weighted average change in the fitness function value, over the specified stall generations is less than the specified function tolerance, then the algorithm is stopped by the Stall generations. The algorithm runs until the function tolerance stops it when the weighted average change in the fitness function value over the specified stall generations is less than the specified function tolerance.

VI. Simulation Results

A. Simulation Setup

The modulation type implemented in this simulation work is the Binary Phase Shift Keying which was at a transmission rate of Mbps at a frame length of ns for the RAKE-GA receiver. There are 1000 symbols in each packet. Spreading is carried out using a ternary code whose length is 24 and at a chip width of 0.167 ns. The UWB multipath channel model as specified in (Foerster, 2003) which does not need channel estimation, considering a single user is used in this simulation. The channel model 3 (CM3), which is a non-line-of-sight (NLOS) environment, at a distance of 4 ~ 10 m, having an average excess delay of 14.18 ns and RMS delay spread of 14.28 ns is the one implemented. L = 10 RAKE fingers is used in the simulation setup.

The population sizes used are P = 20, 40, 50, 60, 80, and 100 with a maximum number of generations being specified as G = 50, 25, 20, 17, 13, and 10 are implemented for the RAKE-GA approach in this work. The fitness limit values are Fl = 0.3, 0.25, 0.2, 0.15, 0.1 and 0.05 and the function tolerance used are Ft = 1e − 6, 1e − 7, 1e − 6, 1e − 8, 1e − 9, and 1e – 10 while the stall generations are StG = 25, 12, 10, 8, 6, and 5. Elite count of 0.05 and crossover of 0.85 was used in this set-up. A value of shrink = 1.0 and scale = 0.75 for the Gaussian mutation and 0.01 is the uniform mutation rate used.

B. Performance Evaluation

The BER performance of the algorithm is presented in Fig. 2 at P = 50 and 100 and maximum generation of G = 20 and 10 for proportional + Gaussian and rank + uniform scenarios with adaptive and fixed parameters respectively. The curve with adaptive and fixed parameters were of the same BER at SNR of 0 − 15dB, despite the fact that population size for adaptive was half of the one used for fixed. Fig. 3 is to compare the performance of rank + uniform, proportional + Gaussian, rank + Gaussian and proportional + uniform, using the same simulation parameters as listed in the previous section. In the figure, rank + uniform and rank + Gaussian are almost of the same BER as the small difference at some population sizes are insignificant. The same applies to proportional + Gaussian and proportional + uniform as we have almost same BER values with insignificant differences at few points. In Fig. 4, we showed the effects of using adaptive parameters on the BER performance of the RAKE-GA for all the four combinations of the two scaling and mutation methods. Fig. 5 shows the population sizes against the two different generations used in the simulation and one obtained after the simulation. We are able to show, that the optimization was terminated in all the cases, with the scenario where Gaussian mutation using more generations than when uniform mutation. All the four scenarios have the same maximum and stall generations as they were specified.

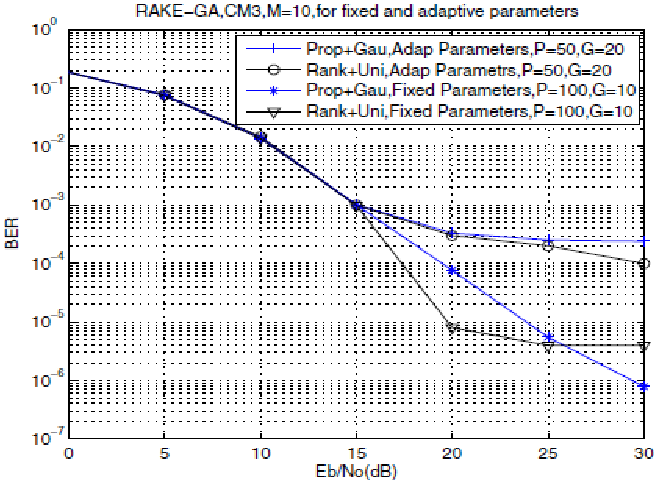


Figure 2: Performance of RAKE-GA IN CM3

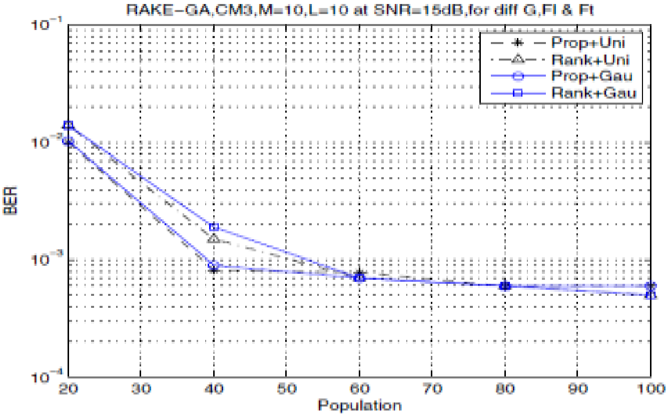


Figure 3: Effect of Population size on Performance of RAKE-GA for CM3

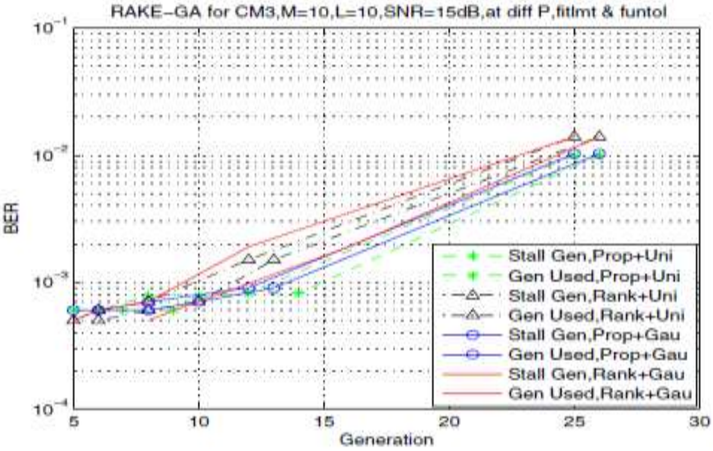


Figure 4: BER vs. Generation for CM3 for RAKE-GA

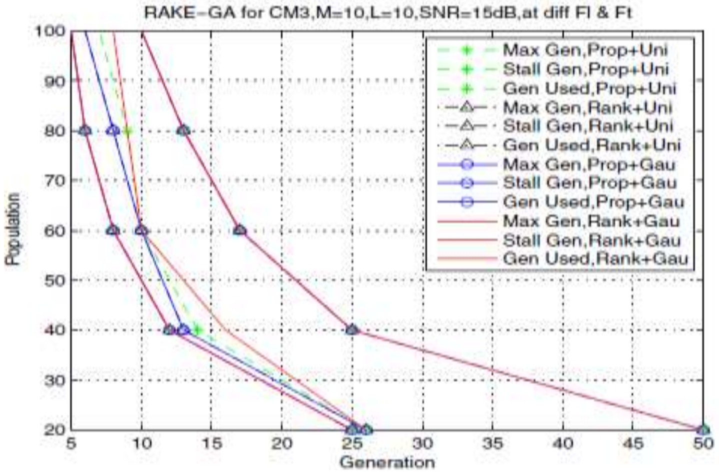


Figure 5: Population size vs. Generation Size for CM3 for RAKE-GA

VII. Conclusion

This research work has been able to conclude that fitness scaling plays very important role in the GA optimization. The mutation on the other hand, has no effect but prevents local convergence. Properly chosen adaptive parameters also improve the performance of the GA. We are able to reach this conclusion from the results of our investigation into the effect of scaling methods and mutation types on the performance of the equalization of the channel for the DS-UWB system using GA as an optimization technique in combination with a RAKE receiver. Two different types of scaling and mutation functions were considered.

References

Perarasi, T. and Ravichandran, V. C. (2014). Optimization of Receivers for UWB Applications. International Journal of Innovative Research in Computer and Communication Engineering, 2(6), pp. 4758 – 4765.

Somayazulu, V. S., Foerster, J. R., and Roy, S. (2002). Design challenges for very high data rate UWB systems. International Conference on Signals, Systems and Computers, (pp. 717–721). IEEE.

Nassar, C. R., Zhu, F. and Wu, Z. (2003). Direct sequence spreading UWB systems: Frequency domain processing for enhanced performance and throughput. International Conference on Communications, (pp. 2180–2186). IEEE.

Sato, H. and Ohtsuki, T. (2005). Computational complexity and performance of RAKE receivers with channel estimation for DS-UWB. Transactions on Fundamentals of Electronics, Communications and Computer Sciences, E88-A(9), pp.2318-2326.

Siriwongpairat, W. and Liu, K. (2008). Ultra-Wideband Communications Systems Multiband OFDM APPROACH. John Wiley & Sons, Inc.

Surajudeen-Bakinde, N., Zhu, X., Gao, J. and Nandi, A. K. (2009). Genetic algorithm based equalization for direct sequence ultra-wideband communications systems. Wireless Communications and Networking Conference. (pp. 1 - 5). IEEE.

Surajudeen-Bakinde, N., Zhu, X., Gao, J. and Nandi, A. K. (2009). Effects of Fitness Scaling and Adaptive Parameters on Genetic algorithm based equalization for DS-UWB. International Conference on Computers and Devices for Communication. (pp. 1 - 4). IEEE.

Montaser, A. M., Mahmoud, K. R., Abdel-Rahman, A. B. and Elmikati, H. A. (2013). Design Bluetooth and Notched-UWB E-Shape Antenna using Optimization Techniques, Progress In Electromagnetics Research, B(47), pp. 279 – 295.

Martins, S. R., Lins, H. W. C. and Silva, C. R. M. (2012).Intelligent Data Engineering and Automated Learning – IDEAL 2012. Springer

Qi, L. Zhengmin, K., Yanjun, F., Jing, Y. and Chen, W. (2011). Multiuser Detection Employing a Novel Genetic Algorithm for UWB Communications. Elsevier Journal on Advanced in Control Engineering and Information Science. 15 (2011), 2505 – 2510.

Kong, Z., Zhong, L., Zhu, G. and Ding, L. (2011). Differential multiuser detection using a novel genetic algorithm for ultra-wideband systems in lognormal fading channel. Journal of Zhejiang University-SCIENCE C (Computers & Electronics), 12(9), 754-765.

Hopgood, A. A. and Mierzejewska, A. (2009). Transform Ranking: a New Method of Fitness Scaling in Genetic Algorithms. In: Research and Development in Intelligent Systems. London, Springer, (349-354).

Gezici, S., Chiang, M., Poor, H. V. and Kobayashi, H. (2005). A genetic algorithm based finger selection scheme for UWB MMSE rake receivers. Proceedings of International Conference on Ultra-Wideband. (pp. 164–169). IEEE.

Wang, M., Yang, S. and Wu, S. (2008). A GA-based UWB pulse waveform design method. Elsevier Journal of Digital Signal Process, 18(1), pp. 65–74.

Hill, S., Newell, J. and O'Riordan, C. (2004). Analysing the effects of combining fitness scaling and inversion in genetic algorithms. Proceedings of International Conference on Tools with Artificial Intelligence. pp. 380–387. IEEE.

Sadjadi, F. A. (2004). Comparison of fitness scaling functions in genetic algorithms with applications to optical processing. Proceedings of SPIE 5557, Optical Information Systems. (pp. 356–364).

Kreinovich, V., Quintana, C. and Fuentes, O. (1993). Genetic algorithms: What fitness scaling is optimal? Cybernetics and Systems, 24, (pp. 9 –26).

Foerster, J. (2003). Channel modelling sub-committee report final. IEEE Technical report. IEEE P802.15-02/490r1-SG3a.

Man, K., Tang, K. and Kwong, S. (1999). Genetic algorithms: concepts and designs. Spriger-Verlag London Limited.

MATLAB (2007). Genetic algorithm and direct search toolbox 2 user's guide, tech. rep., The Mathworks, Incorporation.

Experimental Assessment of Cellular Mobile Performance along the Railway Corridor

Obiyemi, O.O.[1],

Omotoso T.V. [2],

Oguntuase V. A.[3],

Tijani, I.[4]

[1,3]Department of Electrical & Electronic Engineering, Osun State University, Osun State, Nigeria

[1]obiseye.obiyemi@uniosun.edu.ng, [3]vicharde@gmail.com

[2]Physics Department, Covenant University, Ota Ogun State Nigeria
[2]omotosho@covenantuniversity.edu.ng

[4]Laplace Technologies, Nigeria
[4]tijani.laplace@gmail.com

Abstract: With the ongoing rehabilitation of the railway transportation sector in Nigeria, improvement in the quality and reliability of the services deliverable becomes crucial. Reliable railway communication infrastructure guarantees effective operation and also ensures connectivity for security, safety, maintenance and passenger communication. This work describes today's network scenario by assessing current cellular performance as it affects a passenger's experience along the railway corridor. A drive test was conducted on the 6th of July, 2014 between 12:58pm and 07:14pm along the railroad linking Oshogbo and Lagos, Nigeria. The measurement setup consists of four TEMS Mobile Sony Ericsson W995 phones, a Personal Computer, a GPS receiver, and a power bank. The measurement was useful in the assessment of coverage, capacity and Quality of Service (QoS) of four mobile radio networks namely: Airtel, Globacom, Etisalat and MTN in the GSM 900 MHz and 1800 bands. Results reveal that no single mobile network service operator consistently serviced the train throughout the 6-hour trip. Also, the results obtained from the drive test represents a true picture of mobile network condition and can be useful in decision making in several areas - from planning and design through optimization and maintenance of the system, with the goal of maximizing quality, capacity and coverage for all mobile networks for improved service delivery on our railway infrastructure.

Keywords: Railway communication; GSM; Performance of Cellular mobile

I. Introduction

Globally, trains remain a dependable mode of transportation for both freight and passengers. It is safe and has been proven to be relatively more reliable than other means of transportation over the years. Although the on-going railway projects in Nigeria are targeted at increasing capacity and expanding the national railroad net-work, the medium is currently underutilized and patronage dwindles significantly.

In order to improve capacity through the provision of secure, safe and attractive railway systems, it is important to explore the use of a reliable railway communication system. Indeed, such systems will serve as the backbone for a converged railway operation and it should ensure connectivity at a good percentage of operation time.

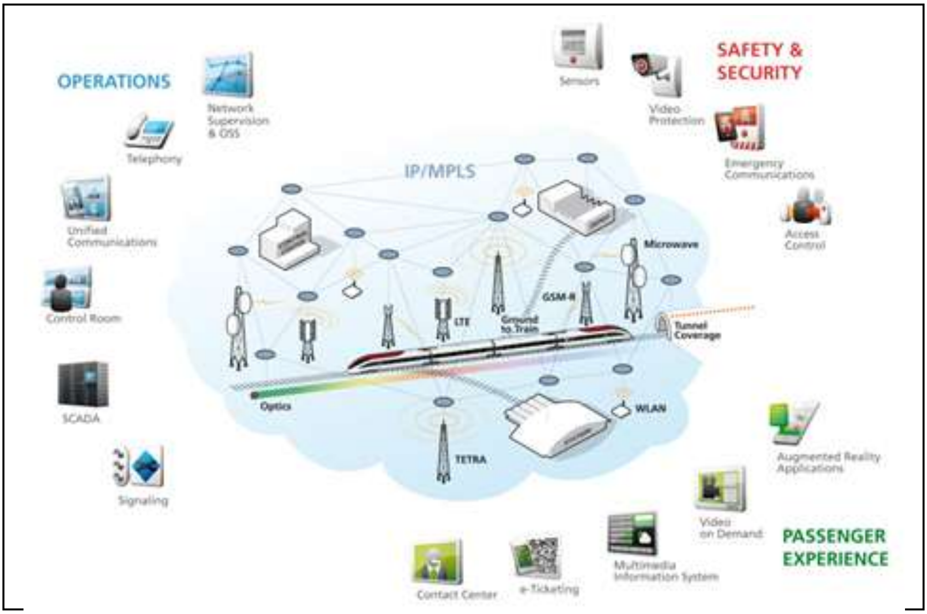


Figure 1. Communication as the backbone for converged rail operations (Bertout & Bernard, 2012)

Interestingly, the evolution of wireless technologies from the 2G centric systems – (e.g. GSM) with limited data transmission capabilities to the 4G broadband multiservice systems (LTE) remains one of the phenomenal advancements in communications in the last decade (Bertout & Bernard, 2012). Typical applications in railway systems as shown in Figure 1 indicates the provision of a two-way continuous communication for safety control, speed control and other vital communication purposes (Aguado et al., 2005; Yan, Chang-Young, Jeong-min, Jin-ho, & Young-Jae, 2013).

Common passenger experience range from the inconsistent waiting time at the train station to network connectivity issues while on the railway corridor. Seemingly, most of the passengers on board are mobile subscribers and they need to enjoy connectivity to their respective operators. Hence, a reliable communication will enhance passenger experience, providing stable voice and data services and retaining connectivity to the diverse social network platforms, which could define the passengers' experience and perhaps compensate for the long hours spent on the railroad. Figure 2 (a) shows existing railway network across Nigeria, while the rail corridor linking Osogbo and Lagos is shown in Figure 2 (b).

This study aims to assess the quality and coverage of the 4 predominant mobile networks on the Lagos-Osogbo railroad. Results obtained provide useful information on the quality of service delivered by MTN, Airtel, Globacom and Etisalat. These results will guide the decisions of mobile network service operators as well as policy formation by the National Communications Commission (NCC), and particularly on the need to adopt dedicated standards such as GSM-R for improved mobile network connectivity along railroads in Nigeria.

II. Related Work

The Global System for Mobile Communication (GSM) standard has been successfully deployed in Nigeria and the performance is frequently assessed in order to guarantee optimal service delivery over respective coverage areas. Interestingly, efforts to sustain connectivity on the highways have yielded positive results and the situation is still improving with the several optimization efforts by the mobile operators.

The situation is however different for the railway corridor. The railway medium of transportation has been abandoned for some time and

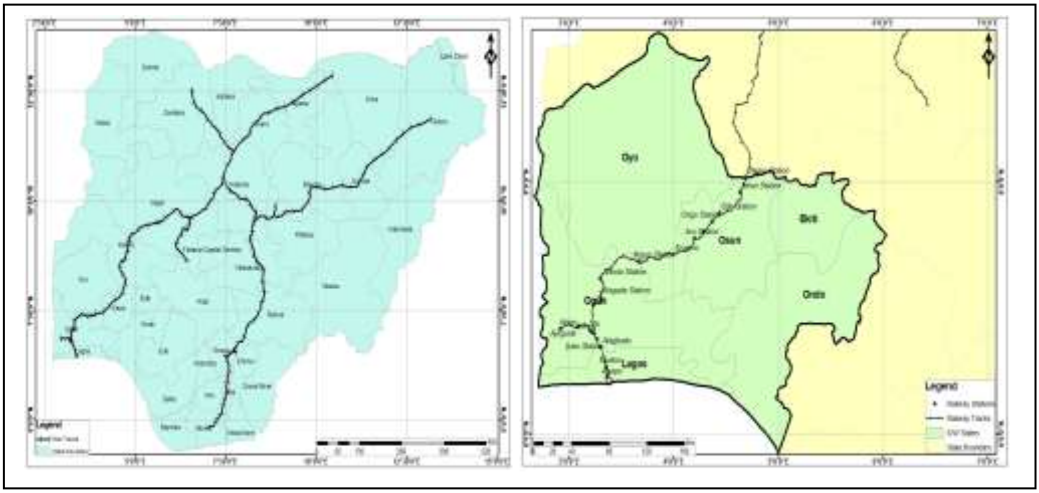


Figure 2. (a) Existing railway network across Nigeria, (b) The railway corridor between Osogbo and Lagos terminal

passengers have lost confidence in the services of the Nigerian Railway Cooperation (NRC). However, patronage has improved recently owing to the ongoing rehabilitation effort by the government. There is the need for the mobile operators and relevant regulatory agencies of the government to work out sustainable train-to-ground communication solutions. Seemingly, GSM remains the main communication infrastructure for both passengers and the train drivers.

Although the performance of GSM has not been evaluated on the railway networks in Nigeria, some related works involving the GSM performance could be mentioned. In (Adegoke & Babalola, 2011; Adegoke, Babalola, & Balogun, 2008), a report on the experiences of mobile subscribers is presented based on the perceived performance of three mobile operators (MTN, GLO and Zain/Airtel) over fourteen states across Nigeria. The quality of service offered was evaluated based on the performance of the selected networks.

Considering the advancement in the telecommunication industry, which is premised on the increase in the number of operators and the subscribers, a review on the performance of the four GSM operators is presented in (Adekitan, 2014). Frequent upgrade and optimization of the existing infrastructure were recommended as solutions to the fickle and unsatisfactory services offered to Nigerians. In a similar approach, the impact of GSM on rural economies in Nigeria was evaluated based on the stakeholders' perceptions (Ajiboye, Adu, & Wojuade, 2007). In their findings, GSM is considered to have sizeable impact on the rural settlements in Nigeria. However, they submitted that the impact on rural dwellers is still marginally poor, hence, confirming the obvious digital divide between the urban and rural settlements. A similar evaluation of the performance of existing GSM services in Nigeria is presented in (Olatokun & Bodunwa, 2006; Popoola, Megbowon, & Adeloye, 2009). In their findings, the limitations of the GSM services were identified and presented, and they are: unreliable QoS, poor network accessibility and retain-ability, and poor network coverage.

In order to enhance services to travelers by train, there is the need to develop an accessible passengers' information system, which can be built on modern networking technologies for the railway. The provision of reliable infrastructure for on-board applications can be built on a mobile telephone network for railways (B3.2, 2001). This of course will also be useful to the train operator(s) on-board, as well as the ground staff. Ultimately, this will also improve operations, safety and the security of the railway system.

Other important train communication solutions have been successfully deployed globally. Some of these standards are demonstrated in

(Bertout & Bernard, 2012), where the Long Term Evolution (LTE) was presented as the next generation of railways and metros wireless communication systems. In their work, the optimal wireless communication system for railways and metro needs were evaluated based on selected performance parameters and service attributes such as voice support, vital traffic, priority, availability, frequency and commercial maturity.

Although GSM-R is presented as the only approved world telecommunication standard for the railway communications (Aldred & Gorasia, 2005; Bibac, 2007; Hofestadt, 1995; Mohamed, 2014), perhaps it has not been deployed on the Nigerian railway system and communication is still based on the conventional GSM for railway communication. However, the need to effectively manage railway operations, ensure passengers' safety and security during their journey, improve travel comfort for passengers, provide real time multimedia information and grant access to social networks in stations or in motion now remains very crucial for modern railway operations.

In this paper, the performance of the four predominant GSM service providers in Nigeria is evaluated experimentally along the busy Osogbo-Lagos railway corridor using five key performance indicators namely; coverage, capacity, accessibility, retain-ability and mobility.

III. Methodology

An experimental measurement campaign was conducted through a drive test which was carried out in a train en route Lagos from Osogbo. The measurement was useful in assessment of the coverage, capacity

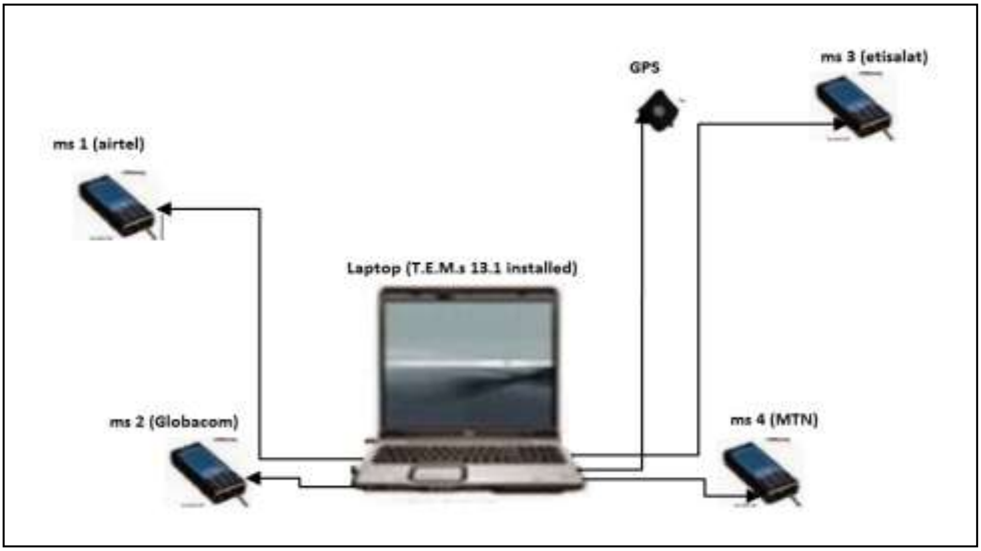
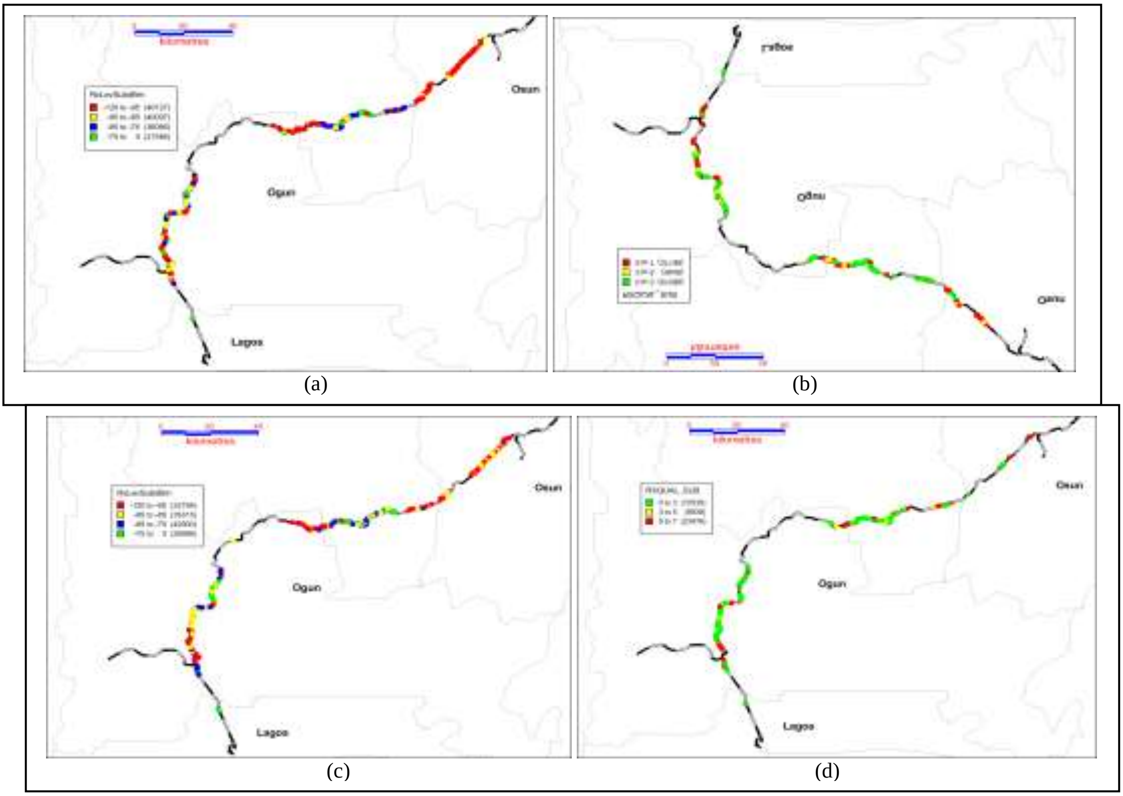


Figure 3. Experimental measurement setup

and QoS of the mobile radio networks.

The measurement setup consists of highly specialized electronic devices, interfaced to mobile handsets. This was used in order to ensure that measurements are realistic and comparable to actual user experiences. The drive test was carried out along the railroad from Oshogbo to Lagos and was conducted on the 6th of July 2014 between the hours of 12:58 Pm to 07:14 Pm. **T**he setup consists of TEMS Mobile Sony Ericsson W995, personal computer (PC) with TEMS Investigation 13.1v installed on it with the TEMS dongle, Global Positioning System (GPS) receiver, power bank. This is as shown in Figure 3.

The drive test equipment was used to collect data relating to the network, especially for services running on the network such as voice or data, radio frequency scanner information and GPS information. The data set collected during the drive testing field measurements include: signal intensity, signal quality, interference, dropped calls, blocked calls, Anomalous events, Call statistics,



(a) (b)

(c) (d)

Figure 4. (a) Rx-lev for Airtel along Osogbo-Lagos rail road, (b) Rx-Qual for Airtel along Osogbo-Lagos rail road, (c) Rx-lev for Globacom along Osogbo-Lagos rail road, (d) Rx-Qual for Globacom along Osogbo-Lagos rail road

Service level statistics, Quality of Service information, Handover information, Neighboring cell information and GPS location co-ordinates.

IV. Results and Discussion

The performance of the four GSM services is described according to different radio frequency measurements namely: Mobile station (MS) Receive-Level (RxLev), MS Receive-Quality (RxQual), MS Speech Quality (SQI). Other Key Performance Indicators (KPIs) for Radio Access Network (RAN) were estimated to assess performance along the railroad. Data collection was possible using TEMs investigation 13.1v and four Mobile Stations (MSs): MS1 – Airtel, MS2 – Globacom, MS3 – Etisalat, MS4 – MTN, all in dedicated mode.

The results obtained were used to present the coverage plot on the route's map using Map Info Professional and the corresponding plots are shown in Figures 5 (a-d).

The KPIs employed in this study are the Call Setup Success Rate (CSSR), Call drop rate (CDR), Handover Success Rate (HOSR), Coverage and Quality.

CSSR is a measure of the MS's accessibility to the GSM network and it is estimated as the ratio of call setup to call attempt (Haider, Zafrullah, & Islam, 2009). Results indicate that MTN records the best accessibility with a CSSR of 84.38% while Airtel had the least with 55.56%. The performance of Globacom was fair with 75.61% while the 83.87% recorded for Etisalat did not do badly, using on this metric.

The CDR is another important KPI, which is a measure of calls that are prematurely disconnected before the end of conversation, against the number of all successfully established calls (Haider et al., 2009). From the results obtained, Airtel records the poorest retain-ability with a large call drop rate of 74.29%,

TABLE 1. STATISTICS FOR THE NETWORK EVENTS FOR THE FOUR MOBILE OPERATORS

Events	MS1 (Airtel)	MS2 (Globacom)	MS3 (Etisalat)	MS4 (MTN)
Blocked Call	21	11	8	9
Call Attempt	63	41	31	32
Call End	19	14	10	13
Call Established	35	31	26	24
Cell Reselection	87	92	66	88
Dropped Call	26	15	13	11
Handover	110	217	299	181
Handover Failure	1	12	10	3

while MTN shows the best retain-ability with a CDR of 45.83%, Globacom (48.39%) and Etisalat (50%) also exhibit an average performance based on this metric. Statistics for the network events for the four mobile operators is presented in Table 1.

The HOSR measures the ability of a customer to talk on the cell phone over a long distance without getting disconnected. It is the ability of a call connection to be handed over from one cell to another without losing the connection (Haider et al., 2009). This KPI is directly linked to call drop rate because a handover failure normally results into a dropped call. The target for this KPI is 90%, meaning only 10% percent of the calls may experience handover failure beyond which the grade of service will decline. All the operators record excellent mobility in this regard. However, Airtel shows the highest mobility with a HOSR of 99.10%, the least is recorded on MTN with 93.78%, Globacom has 94.76%, while Etisalat records 96.76%. This is also shown in Figure 5.

Result obtained on the coverage estimate for the four GSM services is also as presented in Figure 5. It is however indicates that MTN records the highest coverage of 77.61% over the Osogbo-Lagos railway corridor. Airtel has the lowest coverage with 70.89%, while Etisalat records fair coverage with 76.27% and then Globacom, with 74.97% coverage.

Since the quality of the mobile radio network is dependent on its coverage, capacity and frequency allocation, MTN records the best quality throughout the measurement with 78.02% while Airtel records a relatively poor quality with 65.51%.

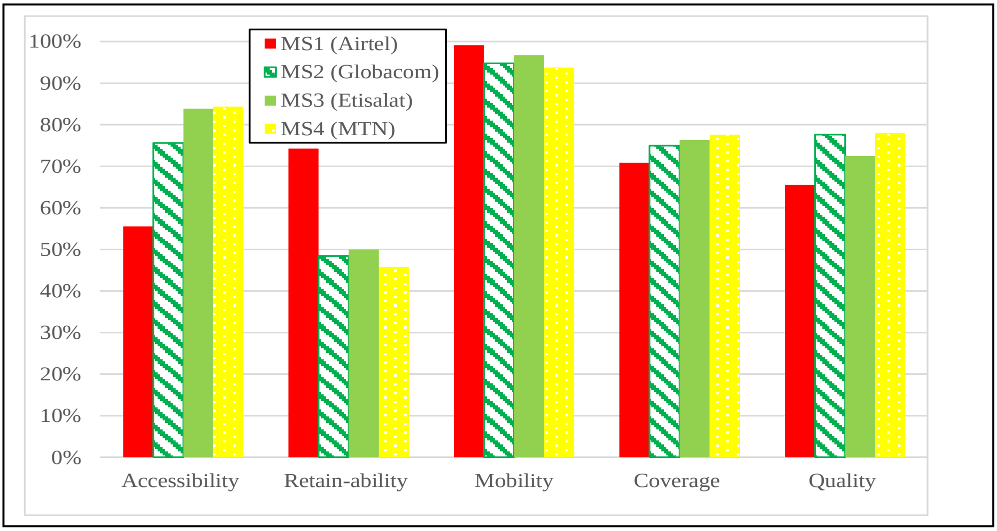


Figure 5. . Key performance indicators for the four mobile operators

Globacom and Etisalat performed averagely well with quality of 77.63% and 72.47% respectively.

V. Conclusions

As patronage increases with the continuous governmental intervention for enhanced capacity, existing railway transport service definitely needs an intelligent train tracking and management system. Moreover, accessibility of railroad travelers to good mobile network could motivate Nigerians to patronize the railway. This is one of the connectivity issues that defines effective operation and maintenance of the entire rail network. There is the need to consider wireless connectivity in the current restructuring and rehabilitation. This therefore calls for the attention of the government, NCC, as well as the mobile network service providers.

The experimental measurement has been conducted on the railroad between Osogbo and Lagos. GSM (2G) coverage and quality along the rail route is fair around locations where Base Transceiver Station (BTS) sites exist. A number of dark spots were observed experimentally as a result of poor coverage gaps.

However, the direct optimization of GSM coverage along the railway routes would be difficult if not impractical as the railway routes are mainly across areas not inhabited by people. Hence the installation of new sites - either 2G or 3G along such routes would not be a good investment for the mobile network service operators concerned.

This situation can be improved by the direct effort of the government to industrialize areas around the railway routes, which would make it reasonable for network operators to install at least a 2G site along such routes as it would serve both the passengers as the train passes and the workers in the industries.

The authors hereby present results based on this experimental campaign. It is considered as a vital tool, which could guide the decision of the mobile network operators and the National Communications Commission (NCC) alike, particularly in the formation of policies in this regard and on the need to adopt dedicated standards such as GSM-R for improved mobile network connectivity along railroads in Nigeria.

Acknowledgements

The authors would like to kindly acknowledge the support of the staff of the Nigerian Railway Corporation, Ibadan and Osogbo offices for granting the team the permission to carry out the measurement - trials and the actual campaign).

References

Adegoke, A. S., & Babalola, I. T. (2011). Quality of service analysis of GSM telephone system in Nigeria. *American Journal of Scientific and Industrial Research, 2*(5), 707-712.

Adegoke, A. S., Babalola, I. T., & Balogun, W. A. (2008). Performance Evaluation of GSM Mobile System in Nigeria. . *The Pacific Journal of Science and Technology, 9*(2), 436-441.

Adekitan, R. A. (2014). Performance Evaluation of Global System for Mobile Telecommunication Networks in Nigeria. *SCSR Journal of Business and Entrepreneurship, 1*(1), 9-21.

Aguado, M., Jacob, E., Saiz, P., Unzilla, J. J., Higuero, M. V., & Matías, J. (2005, 25-28 Sept., 2005). *Railway signaling systems and new trends in wireless data communication.* Paper presented at the Vehicular Technology Conference, 2005. VTC-2005-Fall. 2005 IEEE 62nd.

Ajiboye, J. O., Adu, E., & Wojuade, J. (2007). Stakeholders' perceptions of the impact of GSM on Nigeria rural economy: Implication for an emerging communication industry. *Journal of Information Technology Impact, 7*(2), 131-144.

Aldred, B., & Gorasia, N. (2005). Railway communications systems.

B3.2, T. C. (2001). *RAILWAY TRANSPORT.* Paper presented at the Fourth Framework Programme for Research, Technological Development and Demonstration of the European Union.

Bertout, A., & Bernard, E. (2012). *Next Generation of Railways and Metros Wireless Communication Systems*. Paper presented at the ASPECT 2012, Queen Elizabeth II Conference Centre, Westminster, London, United Kingdom.

Bibac, I. (2007). *GSM-Railway as part of the European Rail Traffic Management System.* Paper presented at the Advanced Topics in Optoelectronics, Microelectronics, and Nanotechnologies III.

Haider, B., Zafrullah, M., & Islam, M. (2009). *Radio frequency optimization & QoS evaluation in operational GSM network.* Paper presented at the World Congress on Engineering and Computer Science.

Hofestadt, H. (1995). GSM-R: global system for mobile radio communications for railways.

Mohamed, H. A. R. (2014). A Proposed Model for Radio Frequency Systems to Tracking Trains via GPS.

Olatokun, M. W., & Bodunwa, I. O. (2006). GSM usage at the University of Ibadan. *Electronic Library, The, 24*(4), 530-547.

Popoola, J. J., Megbowon, I. O., & Adeloye, V. S. A. (2009). Performance evaluation and improvement on quality of service of global system for mobile communications in Nigeria. *Journal of Information Technology Impact, 9*(2), 91-106.

Yan, S., Chang-Young, L., Jeong-min, J., Jin-ho, L., & Young-Jae, H. (2013). *Study on the Effectiveness of High-Speed*

Railway Communication and Signaling System Based on 4G L TE Technology. Paper presented at the 13th International Conference on Control, Automation and Systems (ICCAS 2013), Kimdaejung Convention Center, Gwangju, Korea.

A Review of Metrics and Modeling Techniques in Software Fault Prediction Model Development

Rinkaj Goyal[1],

Pravin Chandra[2],

Yogesh Singh[3]

[1] USICT, Guru Gobind Singh Indraprastha University, Sector 16C, Dwarka, Delhi-110078 [1]rinkajgoyal@gmail.com
[2] USICT, Guru Gobind Singh Indraprastha University, Sector 16C, Dwarka, Delhi-110078
[2] chandra.pravin@gmail.com
[3] USICT, Guru Gobind Singh Indraprastha University, Sector 16C, Dwarka, Delhi-110078[4]
[3]ys66@rediffmail.com

Abstract: This paper surveys different software fault predictions progressed through different data analytic techniques reported in the software engineering literature. This study split in three broad areas; (a) The description of software metrics suites reported and validated in the literature. (b) A brief outline of previous research published in the development of software fault prediction model based on various analytic techniques. This utilizes the taxonomy of analytic techniques while summarizing published research. (c) A review of the advantages of using the combination of metrics. Though, this area is comparatively new and needs more research efforts.

Keywords: Metrics Suite; Object oriented metrics; Software fault prediction; Software Metrics.

1. Introduction

Development of fault prediction models in software engineering is a field more than three decades old; however is still an emerging aspect of empirical software engineering (Catal and Diri, 2007; Kaner and Bond, 2004; Matsumoto et al., 2010; Radjenovic et al., 2013). The resurgence in this field occurs due to availability of public available data as repositories in recent decade and as well as due to the development of other numerical techniques, which have been researched in considerable depth(Dick et al., 2004).

A fault prediction model uses statistical methods to assess and quantify the relationship between different metrics and fault-proneness of a software module even before it is released (Catal and Diri, 2009; Catal et al., 2011; Hall et al., 2011; Raj Kiran and Ravi, 2008).

Different object-oriented metrics have been proposed in the literature due to the increased usage of object-oriented technology in software development(Aggarwal et al., 2009, 2006; Anh, 2010; Arisholm et al., 2010; Babic, 2012; Caglayan et al., 2010; Catal, 2011; Chowdhury and Zulkernine, 2011).

Predictive models quantitatively estimate some aspect of system quality and their efficiency is determined by fault history data and applied quality evaluation procedures(Corazza et al., 2010; Couto et al., 2012; Hong et al., 2010; Janes et al., 2006; Jones, 2008; Khoshgoftaar et al., 2006; Lavazza and Robiolo, 2010; Li and Henry, 1993; Li et al., 1991; Luo et al., 2010) Object oriented development needs a different strategy towards the development of metrics. Since, object-oriented technology utilize objects as its building blocks and contrasting from procedural systems, which use algorithms instead. The derivation and consequently the selection of appropriate metrics depend on the identification of attributes of objects and peculiarities of object-oriented software development process. These metrics not only indicate the complexity of an object and its association (interaction) with other objects, but also measure different characteristics of a quality model **(Figure .**Error! Reference source not found.**).**

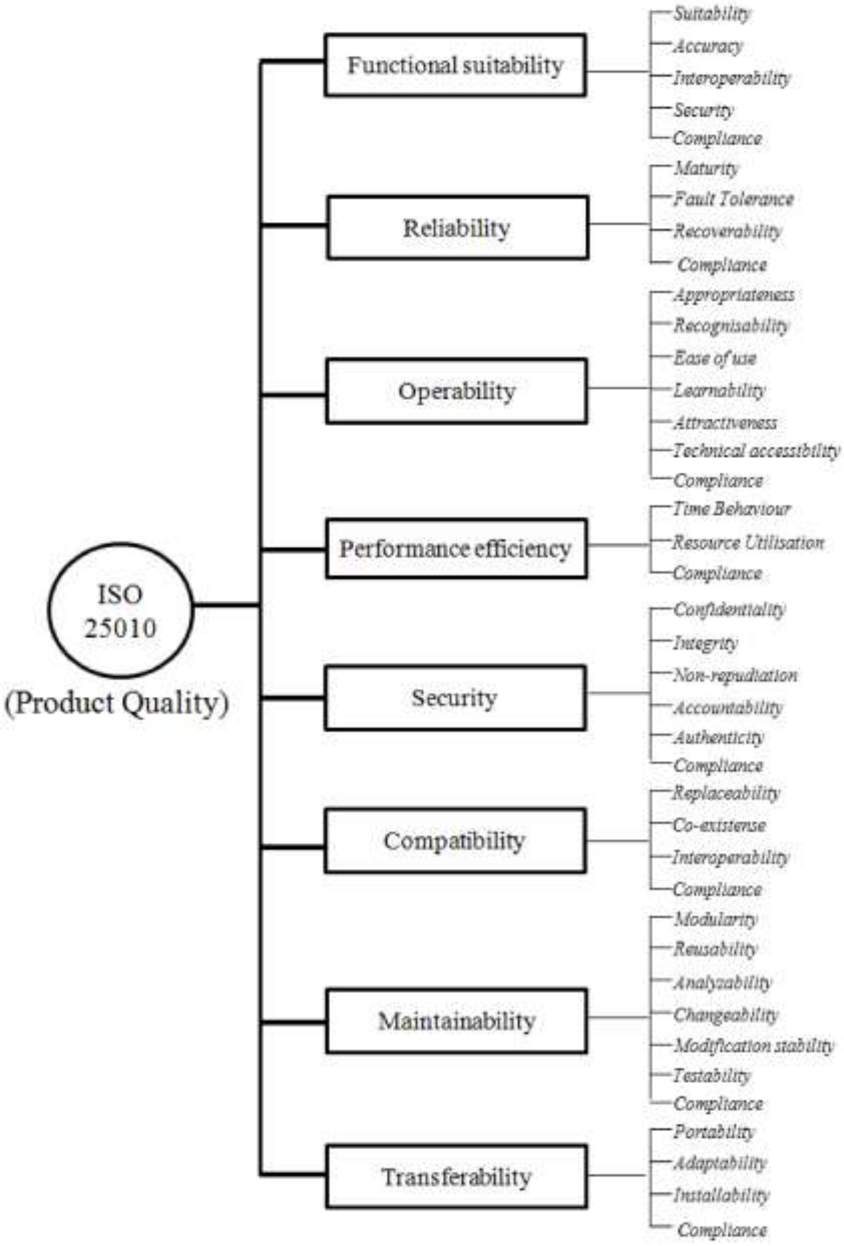


Figure 1: ISO/IEC 25010 Software Quality Standard (Adapted from Wagner et al. (Wagner, 2013)

2 Software Metrics and Suites: A Survey

Software metrics can be categorized as product metrics, process metrics or resource metrics. Product metrics measure different features of developed programs like Methods and Class level metrics in object-oriented systems. Process metrics are related to the measurement and quantification of activities like design, implementation, testing, and maintenance. Resource metrics focus on all other resources involved in development such as programmers, cost of the product and processes, etc. (Ebert and Dumke, 2007; Koru and Liu, 2005; Laird and Brennan, 2006; Lanubile and Visaggio, 1997).

These metrics have shown a corresponding relationship with a variety of external quality characteristics of software, such as reliability, testability and maintainability (Alshayeb and Li, 2003; Li and Henry, 1993; Mair and Shepperd, 2011).

Carapuça et al. (Carapucca and Others, 1994) suggested a classification skeleton that represents the taxonomy of Object oriented metrics. This framework is known as TAPROOT ((Taxonomy Précis for Object-Oriented Metrics) portrayed as a tubular arrangement with two independent vectors (Figure 2); different aspects of measurement (design, size, complexity, reuse, productivity, quality) and the granularity (method, class, system) of an object-oriented system. Though, there are no obvious boundaries between different categories and overlapping may be observed. However, this framework promotes the necessity of relevant metrics adequately to address a particular dimension of the software module. Figure 3 sketches the OO design measures apprehending varying dimensions and architectural quality of a class identified by Briand et al. (Briand et al., 1998). These measures associate to the fault-proneness of a class.

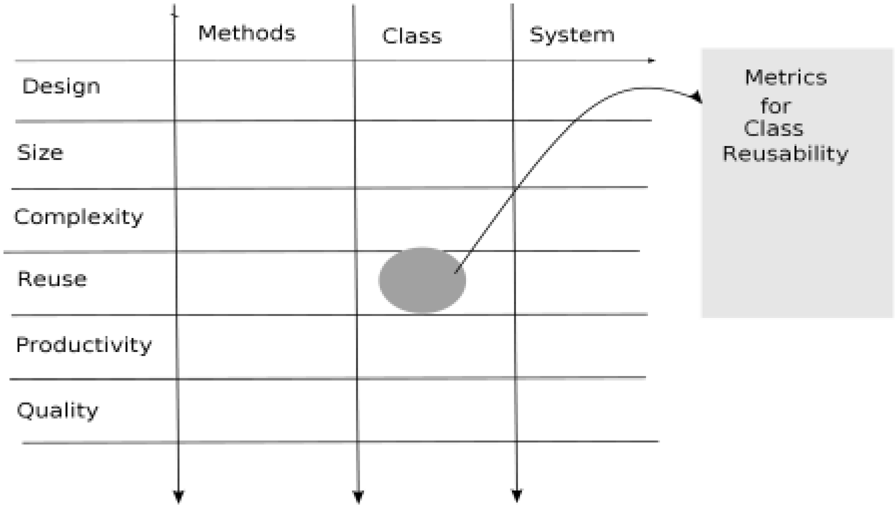


Figure 6: Taxonomy for Object-Oriented Metrics.

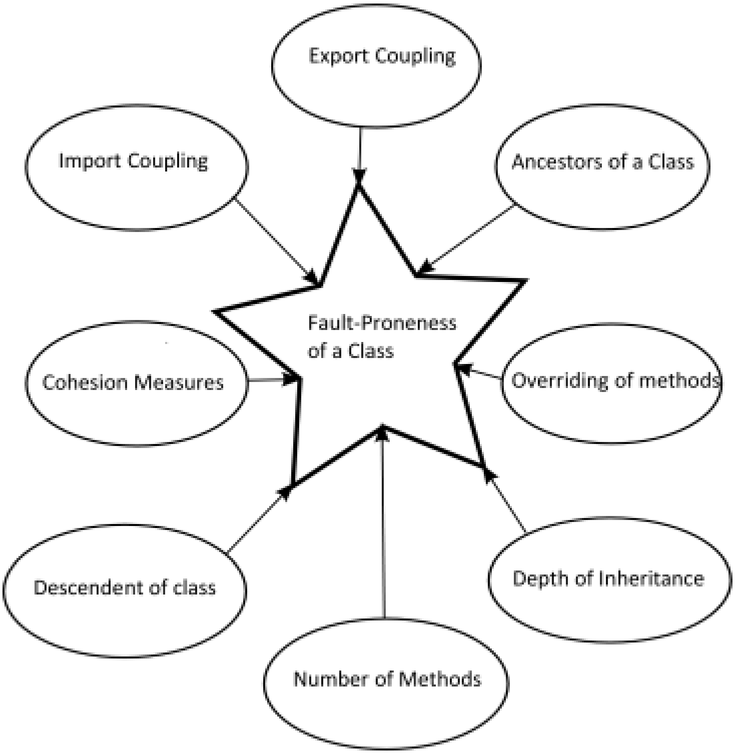


Figure 7: OO Design Measures Related to Fault-Proneness.

A concise summary of the development of different metrics (metrics suite) in the chronological order of their reporting is given below;

McCabe T. (McCabe, 1976) proposed a graph-theoretic measure to compute program’s structural complexity known as Cyclomatic Complexity (CC). When a program is modelled as a control flow graph, CC is defined as follows:

$$CC=e-n+p \quad (1)$$

Where n = number of vertices; e = number of edges and p = connected components.

Conceptualising coupling as the critical complexity measure for fault prediction, Li, W. and Henry Li (Li and Henry, 1993) suggested two metrics Message Passing Coupling (MPC) and Data Abstraction Coupling (DAC), based on the coupling through message passing moreover, Abstract data types (ADT) declared in the class. Two size metrics i.e. SIZE 1 and SIZE 2 were also recommended, addressing the ambiguity in the determination of the size factor of an object-oriented program

(Table 1).

TABLE 2: METRICS PROPOSED BY LI AND HENRY (Li et al., 1991)

Metric Name	Description	Category
MPC(Message Passing Coupling)	Number of send statements defined in a class	Methods Design
Number of Methods (NOM)	Number of local methods	Method Complexity
SIZE1	Number of semicolons in a class	Attribute size
SIZE2	Number of attributes + Number of local methods	Attribute size

Chidamber, S. R. and Kemerer (Chidamber and Kemerer, 1994) proposed an extensively applied and validated metrics suite, commonly identified as Chidamber & Kemerer (CK) metrics suite with six metrics (Table 2).

Hitz, M. and Montazeri (Hitz and Montazeri, 1995) discussed flaws in the determination of coupling constituent in CK metrics suite. They proposed two coupling based metrics, Coupling among objects(CLO) and Coupling among classes (CLC) by analysing the coupling between classes and object as two distinct impressions (Table 3).

TABLE 3: METRICS PROPOSED BY CHIDAMBER & KEMERER (Chidamber and Kemerer, 1994)

CK Metric	Description	Category
Coupling Between Object classes (CBO)	represents the dependence of one class over other classes	System Complexity
Depth of the Inheritance Tree (DIT)	represents the length of the longest path from a given class to the root class in the inheritance tree	Class Design
Lack of Cohesion Metric (LCOM)	represents the count of method pairs in a class with zero similarities	Class Design/Method Complexity
Response for the classes (RFC)	represents the sum of the number of local	Class Design

	methods and remote methods	
Weighted Methods per Class (WMC)	represents the sum of the complexity of methods.	Method Complexity
Number of Children (NOC)	represents the count of the number of immediate subclasses of a class.	Class Complexity

TABLE 4:METRICS PROPOSED BY HITZ & MONTAZERI

Metric Name	Description	Category
CLO(Coupling among objects)	Represents dynamic dependencies between objects	System complexity
CLC(Coupling among classes)	Represents static dependencies between implementations	System complexity

Tegarden et al. (Tegarden et al., 1995) introduced following metrics through verifying that interaction and inheritance are the determining factors in the coupling aspect of a class. Whereas, features like association and generalization-specialization contributes towards the cohesiveness (Table 4).

Abreu et al. (e Abreu and Melo, 1996) proposed a MOOD (Metrics for Object Oriented Design) metrics suite comprising of the metrics listed in Table 5. These metrics capture core architectural ingredients of an object-oriented program like encapsulation, inheritance, polymorphism and message passing. Bansiya et al. (Bansiya and Davis, 2002) proposed QMOOD (Quality model for object-oriented design) metrics suite with an assessment of total quality index as super metric. These eleven metrics are based on the design quality attributes defined in ISO 9126 and possess an edge of early computability in the design process (Table 5).

TABLE 4: METRICS PROPOSED BY TEGARDEN, D. P et. al. (Tegarden et al., 1995)

Metric Name	Description	Category
CLD(Class-to-leaf depth)	Count the maximum levels that are below the class in the inheritance hierarchy	Class complexity
NOA(Number of ancestors)	Count of the parent classes of the class.	Class complexity
NOD(Number of descendants)	Count of the descendent classes of a class.	Class reusability

TABLE 5: METRICS PROPOSED BY ABREU et. al. (Abreu and Melo, 1996)

Metric Name	Description	Category
Method Hiding Factor (MHF)	Average (in per cent) of the methods visibility.	Class design
Attribute Hiding Factor (AHF)	Average (in per cent) of the attribute visibility.	Class design
Method Inheritance Factor (MIF)	Average (in per cent) of methods reusability.	Method reusability
Attribute Inheritance Factor (AIF)	Average (in per cent) of attributes reusability.	Class reusability
Coupling Factor (COF)	Average (in per cent) of class coupling.	Class complexity
Polymorphism Factor (POF)	Average (in per cent) of methods overridden.	Method complexity

Software measurement research community is actively involved in identifying new OO metrics addressing more quality attributes of Object-oriented software. Recent work in this regards includes the following;

Michura et al. (Michura et al., 2013) proposed complexity metrics to determine the difficulty in implementing changes through the measurement of a method's complexity, diversity, and complexity density (Table 6).

Wang et al.(Wang and Shao, 2003) proposed Cognitive complexity as a new measure to determine the complexity by taking the cognitive and psychological parameters into account.

These parameters consider internal structures of the artifact along with the processed input-output into consideration to measure particular facet of the quality of a software. Misra et al. (Misra and Adewumi, 2014; Misra, 2011; Misra et al., 2012)proposed following cognition driven complexity measures (Table 7).

TABLE 6:METRICS PROPOSED BY MICHURA et al. (Michura et al., 2013)

Metric Name	Description
Mean Method Complexity (MMC)	Measures the complexity of a class method obtained by dividing method's cyclomatic complexity with the number of methods in a class.
Standard Deviation Method Complexity (SDMC)	measure the method diversity of a class by taking the deviation of a methods complexity from the mean of methods complexity into consideration.
Proportion of Nontrivial Complexity (PNC);	measures the class complexity density by identifying the proportion of methods whose complexity is not one.

TABLE 6: METRICS PROPOSED BY MISRA et al. (Misra et al., 2012)

Metric Name	Description
Method Complexity (MC)	Measures the complexity of a method by taking logical structures used in a method into consideration. This metric is computed by assigning a weight to each logical structure involved in the implementation of a method followed by summing up the complexity thus obtained for all methods of a class.
Coupling weight for a class (CWC)	Measures the coupling effect between classes by not only considering the number of messaged passed, but also taking the complexity of calling and called functions into consideration.
Attribute Complexity (AC)	Measures the complexity induced in the class due to data members of a class. This metric is obtained by summing up the number of attributes.
Weighted Class Complexity (WCC)	Measures the class complexity as a whole by summing up the methods and attributes complexity.
Code Complexity (CC)	Measures the complexity introduced due to inheritance by differentiating between the influence of sibling and child-parent relationship in the determination of the overall impact.

3 Review of Modeling Techniques

In recent years empirical software engineering has seen an increased usage of various data analytic techniques accruing to the public availability of a multitude of software repositories (Harrison et al., 1998; Mende, 2010; Menzies et al., 2010; Mertik et al., 2006; Perry et al., 2000; Rodriguez et al., 2012; Runeson et al., 2006; Seaman, 1999; Shepard et al., 2001) and progressive research shown by machine learning and data mining community. Nonparametric techniques like Regression Tree, Random Forest, Support Vector Machine, Neural Network etc. have been extensively reported tang(Brady and Menzies, 2010; Lessmann et al., 2008; Malhotra et al., 2010; Succi et al., 2003; Tang et al., 1999; Tichy, 1998). Following is the review of fault prediction model's evolution based on the grounds of applied data analysis routines. Fig. 4 outlines the categorization of the analysis methods studied in this section along with their position in the hierarchy of the broad spectrum of data science.

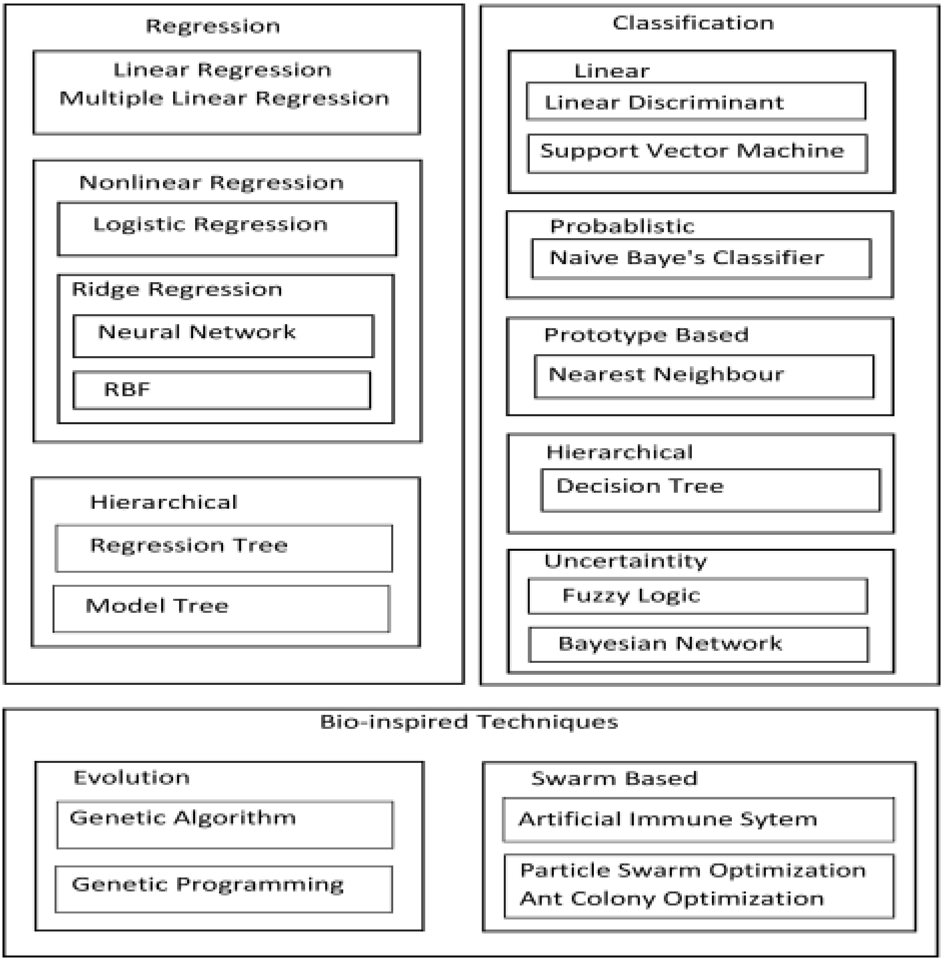


Figure 8:Taxonomy of Data Analysis Techniques

3.1 Linear and Logistic Regression

A statistical method for regression analysis is widely reported technique to construct fault prediction models.

In multiple linear regression (MLR) technique, relationship between two or more independent variables (x_1, x_2 ...x_k) with a dependent variable (y) is determined. The developed model can be viewed as Data = Fit + Residual.

To fit the model (i.e. to find regression coefficients) the ordinary least square method (OLS) is performed minimizing the squared distance between predicted and actual values, and the value of the relationship computed by the model can be predicted from residuals (Uysal and Guvenir, 1999; Yan and Su, 2009).

Logistic regression works like linear regression, except for the fact that independent variables may be categorical, and the response is a dichotomous outcome ranging from 0 to 1(Runkler, 2012).

Alshayeb and Li (Alshayeb and Li, 2003) established the relationship between OO metrics selected from Chidamber and Kemerer (CK) metrics suite (Chidamber and Kemerer, 1994) and development/maintenance efforts like Lines Changed (LC), Lines Added (LA), and Lines Deleted (LD) using Multiple Linear Regression (MLR). Even so, such a relationship was limited to short-cycled agile process and was found ineffective in the long-cycled framework process..

Basili *et al.* basili(Basili et al., 1996) used logistic regression to analyse the relationship between OO metrics and fault-proneness of classes during the early phases of the life-cycle. They had evaluated each metric in isolation using the univariate method, augmented by multi-variate regression to evaluate the predictive capability of those metrics. The outcomes of this study were validated with the data gathered from eight medium-sized software modules developed in C++.

Briand *et al.* (Briand et al., 2000) used logistic regression to the subset of OO metrics in the development of fault prediction model, owing to the fact that many OO metrics capture similar dimensions of measurement. The investigations were made with 28 coupling measures, ten cohesion measures, and 11 inheritance measures. Their work concluded the prevalence of coupling and inheritance measures and cohesion measures were found ineffectual.

Emam *et al.* (El Emam et al., 2001) illustrated the impact of confounding effect of class size in validation studies using logistic regression. Their study considered CK metrics and a subset of the Lorenz and Kidd metrics (Lorenz and Kidd, 1994) for a large C++ telecommunications framework. Supporting their argument, they suggested an Export Coupling (EC) metric and statistically established its strong association with fault-proneness.

Marcus *et al.* (Marcus et al., 2008) employed logistic regression accompanied by principal component analysis (PCA) on three open source software systems in support of their new measure for class cohesion: Conceptual Cohesion of Classes (C3). Their work concluded the superiority of the C3 metric over existing structural metrics.

3.2 SVM and Instance-based Learning

A Support Vector Machine (SVM) optimally separates data points into two categories using a kernel function. Model thus developed using an appropriate kernel function is closely related to the neural network and generalize well, though starting with a small training sample. This engenders SVM a suitable technique to develop fault prediction models, where the information of complexity metrics in the early phase of SDLC is very limited.

Elish *et al.* el(Elish and Elish, 2008) measured performance of Support vector machine (SVM) in classifying faults-prone software modules employing four publicly available NASA data sets. These data sets were derived from software projects developed in the different programming languages (C, C++, and Java). The predictive accuracy of the models developed through SVM with 21 static module level metrics with 10 fold cross validation was compared against eight other statistical and machine learning techniques (LR, KNN, RBF, MLP, NB, BBN, RF, and DT)[1] SVM showed superior performance of recall measures whilst also maintaining significant high values of F-measures.

Xing *et al.* (Xing et al., 2005) explored the utilization of SVM and its extended form (transductive SVM i.e. TSVM) on a random sample of 390(40000 lines of code) routines of a medical imaging software developed in Pascal, FORTRAN, assembly, and PL/M. A total of eleven complexity metrics was considered for model development. When compared to Quadratic discriminant analysis (QDA) as a classifier blended with PCA as the feature selection technique, SVM with RBF as the kernel trick based classifiers were reported to result in improved classification accuracy measures.

Di Martino *et al.* d(Di Martino et al., 2011) confirmed the advantages of using SVM as the linear classifier. However, they reasoned the applicability of SVM for non-linear classification. The parameters of underlying kernel function ought to be tuned by using the statistics of dataset. For example, in case of RBF as the kernel function,

[1] LR : Logistic regression, KNN: K-nearest neighbour, RBF: Radial basis function, MLP: Multi-layer perceptron, BBN: Bayesian belief network, NB: Naive Bayes, RF: Random forest, DT: Decision tree.

parameters like C (penalty factor for misclassified points) and γ (radius of the RBF) have an impact on classification accuracy. They recommended a genetic algorithm (GA) based approach to tune these parameters optimally for the dataset of Jedit software module available in the PROMISE data repository. The conclusion derived make evident the higher performance of the SVM models combined with GA.

3.3 Bayesian and Evidence-based Statistics

A Bayesian network (BN) represents an acyclic graph that embodies the joint probability distribution of a set of random variables. It models the casual influences on the problem and has not been explored in depth in the software measurement field, particularly in the predictive analytics of fault prediction model development. Construction of BN requires the modeling of qualitative influences in a domain through graphs and after that assignment of probabilities to each node in the representation.

Pai *et al.* (Pai and Dugan, 2007) developed BN by taking all products, process and another source of information accounting for fault introduction in software into consideration. Mining of product and process metrics data generates an individual BN structure. These different BN structures estimate external quality metrics like Fault content, Fault Proneness, reliability, etc. to predict the overall quality of software. This study summarizes contradictory, but interesting results. Significance of WMC, CBO, RFC, and SLOC metrics, with MLR as the mechanism to construct BN, supports the results reported by Gyimothy et. al. (Gyimothy et al., 2005) and insignificant metrics include DIT and NOC metrics.

Fenton *et al.* (Fenton et al., 2002) developed a toolkit AgeneRisk (available at http://www.agenarisk.com) to generate a dynamic Bayesian network that allows the construction of causal models to any phase of software Life cycle. The utilization of toolkit exhibited significantly improved and validated predictive accuracy in a trail of 30 different projects.

Bai *et al.* (Bai et al., 2005) developed a Markov Bayesian Network (MBN) to incorporate dynamic change in the model parameters of BN. To develop MBN, core ingredients shown are; initial distribution of defects computed from the data set, distribution of failure time and distribution of the number of defects removed over time. Their results concluded enhanced performance compared to traditional JelinskiMoranda model (JM model) and GoelOkumoto NHPP model (GO model).

Dejaeger *et al.* (Dejaeger et al., 2012) studied 15 different Bayesian Network (BN) classifiers using

NASA and Eclipse foundation data set and inferred that a general Bayesian network can be outperformed by the naive Bayes classifier when expanded with different augmentation operators like Tree augmenter, Forest augmenter, and selectively augment with and without discarding.

3.4 Additive Models and Trees

A classification and regression tree (CART) is a treelike representation of a succession of decisions involved. Each internal node encapsulates a decision taken to carry out subsequent predictions(Death and Fabricius, 2000; Dvzeroski and Drumm, 2003). In a classification tree (decision tree), labels are associated with the leaves, whereas, in the regression tree, the actual numerical value of the response variable is assigned to the leaf (Breiman et al., 1993). Model trees are an extension of regression trees that unite a linear model with each of the leaves instead of merely a numerical value (Frank et al., 1998; Quinlan, 1992).

The regression tree model for fault prediction was first reported by Gokhale and Lyu (Gokhale and Lyu, 1997). Since then a large number of studies have used these trees-based regression techniques, relevant amongst them are following:

Khoshgoftaar *et al.* kho(Khoshgoftaar et al., 2002) illustrated the effectiveness of a regression tree algorithm to identify fault-prone modules for 4 consecutive releases of a large telecommunications system using 24 product and four execution metrics.

Bibi *et al.* (Bibi et al., 2008) performed regression via classification (RvC) by discretizing target variables to train the classification model, and then reversed the process to change the output, back into a numerical prediction.

In this study, they experimented with different classification algorithms viz IBk JRip, PART, J48, and SMO available in Weka environment (Witten and Frank, 2005) using Pekka data set of a commercial bank (Maxwell, 2002) to validate the superiority of RvC approach.

Guo *et al.* guo(Guo et al., 2004) statistically analysed the relative performance of random forest over logistic regression and discriminant analysis using five case studies on a NASA data set. Random forests are variations of the decision trees and in this study, they generate a large number of such trees with the training data to establish the preponderance of random forest empirically.

Chowdhury *et al.* (Chowdhury and Zulkernine, 2011) analysed techniques like C4.5 Decision Tree, random forests, and logistic regression. They used fifty-two releases of Mozilla Firefox, developed over a period of four years to compare predictive performances. Their study concluded that the majority of the vulnerability-prone files in Mozilla Firefox can be identified with these techniques well within the tolerable false positive rates.

3.5 Perceptron based Models

Neural networks are universal approximation category of nonlinear regression method based on the action of biological neurons. In general, the term "Neural Network" (NN) and "Artificial Neural Network" (ANN) belongs to a Multilayer Perceptron Network. Additional prototypes of neural network include Probabilistic Neural Networks (PNN), General Regression Neural Networks (GRNN), Ward neural network (WNN), Radial Basis Function (RBF), Recurrent Networks and Hybrid Networks etc (Yuhas and Ansari, 2012)[**Error! Reference source not found.**].

Zheng *et al.* (Zheng, 2010) took the severity of type II error into consideration to develop neural network-based predictive models. Type II error deals with the misclassification of defect-prone modules, whereas Type I error relates the misclassification of not-defect-prone ones. Neural Network with cost-sensitive Adaboost (boosting technique) (Runkler, 2012) manifested reduced number of such type II errors.

Khoshgoftaar *et al.* (Khoshgoftaar et al., 1997) first illustrated the utilisation of neural-network for EMERALD (Enhanced measurement for early risk assessment of latent defects), a joint project of Nortel and Bell Canada to improve the reliability of software. Their results manifested that neural manages Type II classification error efficiently compared to discriminant analyses.

Kanmani *et al.* (Kanmani et al., 2007) compared and analysed the performance of Back Propagation Neural Network (BPN) and Probabilistic Neural Network (PNN) to predict the fault-proneness of the C++ modules with conventional logistic regression using the data set generated from the software modules developed by the graduate students. This study empirically verified the robustness of the predictive accuracy of PNN using five quality parameters.

Thwin *et al.* (Thwin and Quah, 2003) analysed the comparative performance of ward neural network (WNN) and General Regression Neural network (GRNN) to predict count of defects in a class and the number of lines change per class. A WNN is a back propagation network with three slabs in the hidden layer having different activation functions. GRNN is one-pass learning and memory based network structure. This study reasoned the superior predictive ability of GRNN over compared to WNN.

3.6 Fuzzy Logic based Approaches

Fuzzy based models change the subjective knowledge into mathematically explorable terms and rules to create systems with a level of uncertainty.

The use of fuzzy logic in the modeling of various perspectives of software development process is increasingly achieving attention of researchers. Following is the concise summary of related contributions published in the literature;

So *et al.* (So et al., 2002) empirically analyzed the performance of fuzzy logic to predict fault-prone modules using inspection data. They built up an automated and scalable system that performs well, even if huge inspection data is not usable.

Pandey *et al.* (Pandey and Goyal, 2009) explored the effectiveness of fuzzy expert system in the prediction of the occurrence of faults after each phase of the software development life cycle (SDLC). Fuzzy inference system of their model employs eight reliability metrics collected for different phases of SDLC.

Xu *et al.* (Xu et al., 2008) demonstrated the inference ability of fuzzy expert system with limited facts available. Their study resulted in the maturation of a risk assessment framework following NASA standards.

Yang *et al.* (Yang et al., 2007) proposed a hybrid model of Neural and Fuzzy logic. This plan uses the knowledge derived from previous similar projects for training and efficiently deals with the data that is objective in nature.

Muzaffar *et al.* (Muzaffar and Ahmed, 2010) analysed the impact of de-fuzzification and membership functions in the conception of a fuzzy logic based system for software development effort.

Verma *et al.* (Verma and Sharma, 2010) proposed a fuzzy logic-based framework for development effort evaluation and reported increased performance on an artificial and live project data both. Their conclusions statistically establish the efficacy of fuzzy logic based system to manage the imprecision in the input data.

Aljahdali *et al.* (Aljahdali and Sheta, 2011) reported encouraging outcomes using fuzzy nonlinear regression in modelling accumulated faults in software modules.

3.7 Bio-inspired Techniques

Evolutionary techniques are bio-inspired meta-heuristic approaches and exhibit common characteristics (Back et al., 1997).

1. Execution of these techniques begins with a population of the candidate solution set constituting the search space.
2. A selection process identifies better solution through a derived fitness criteria depending upon the problem formulation.
3. New solutions evolve through mutation and recombination.

Azar *et al.* (Azar and Vybihal, 2011) optimized existing software quality estimation models using ant colony optimization (ACO) technique. ACO adapted with previously developed predictive models put to use a common domain and context-specific data for model construction. This permits to infer predictive models built for one dataset for new data. The result of this study concluded with the enhanced performance of ACO compared to C4.5 and random guessing techniques.

Khoshgoftaar *et al.* (Khoshgoftaar and Seliya, 2003) investigated the influence of genetic programming (GP) in developing decision trees to solve software quality classification problem whilst minimizing the cost of misclassification and the size of tree simultaneously. Two initial releases of large windows based embedded systems comprising of more than 27 million lines of codes generated dataset used in this study. The results concluded that GP based decision tree modelling accounts for greater flexibility in building optimal classification models.

Vandecruys *et al.* Vandecruys (Vandecruys et al., 2008) empirically verified the advantage of AntMiner+ classification process over C4.5, logistic regression and support vector machines using NASA data repository to predict faults in the software module. AntMiner+ is a classification method based on ACO and deduces a rule-based classification models from a dataset. The Implementation of AntMiner+ is accessible on the web (Refer http://www.antminerplus.com).

Bouktif *et al.* (Bouktif et al., 2010) trained predictive model parameters from already built models. In the proposed mechanism, new models develop through the genetic algorithm based combination and adaptation of the expertise already available in existing prediction models. The application of this mechanism with decision trees over NASA data achieved significantly improved selection of models.

Chiu *et al.* (Chiu, 2011) in one way extends the previous work of Bouktif et. al. [**Error! Reference source not found.**] and suggested an integrated decision network (IDN) wherein particle swarm optimisation (PSO) implements the combination and adaptation phases of the model development. In comparison to GA, PSO approach needs fewer complex operators, hence makes it more appropriate to design IDN. The derived results establish that the proposed mechanism outperforms individual software quality classification models and provides a deeper insight to decision makers.

Nature inspired computational techniques like the Artificial Immune system have been used in fault prediction and performance and are reportedly better than J48 classifiers (Catal and Diri, 2007). Search-based software engineering (SBSE), which utilizes nature-inspired techniques in empirical software engineering is an emerging field.

SBSE is gaining momentum with the advent of enhanced heuristic algorithms (Gay, 2010; Harman, 2010; Harman et al., 2012, 2009; Meziane and Vadera, 2010).

Studies indicated below points to the investigations, which take advantage of the combination of some of the techniques above and address other relevant aspects of software measurement:

Bibi *et al.* (Bibi et al., 2008) used a combination of classification and regression techniques by executing regression, via classification. Gyimothy *et al.* (Gyimothy et al., 2005) validated metrics for fault-proneness predictions in the "Bugzilla" database using a combination of regression and machine learning methods.

Nagappan *et al.* (Nagappan et al., 2006) provided an excellent step by step guide to develop quality predictors.

Beecham *et al.* (Beecham et al., 2008, 2006) and Kitchenham et.al. (Kitchenham et al., 2009, 2002) provide with notable systematic literature reviews (SLR) in empirical software engineering, along with an unfolded mechanism to administer a new, although other suitable literature reviews are also accessible (Biolchini et al., 2005; Petersen et al., 2008).

Menzies and Shepperd (Menzies and Shepperd, 2012) express their opinions about the sample size, applied statistical techniques and the conclusion stability of the published results in the editorial of the "*Special issue on repeatable results in software engineering prediction*". This premium editorial give emphasis on the reproducibility of the published results and infers the studies made by Dybaa *et al.* (Dybaa et al., 2006) and Easterbrook *et al.* (Easterbrook et al., 2008). Further, Singer *et al.* (Singer and Vinson, 2002) recognizes ethical and legal issues implicated in empirical software engineering.

4. Fault Prediction Using Metrics Combination

The Software Development Life Cycle transforms artifacts like a software requirement specification (SRS) to a final product. The nature of the relationship between artifacts and suitable transformation leads to a large number of the resultant artifacts (Raffo et al., 2000). Combination of metrics, selected from different phases of the software development lifecycle, results in improved accuracy of predictive models.

However, while combining several metrics; the issue of multi–collinearity arises due to inter-correlation among the metrics. To overcome this, various feature selection techniques like Principal Components Analysis (PCA) may be used. With PCA, a smaller number of uncorrelated linear combinations of metrics can be obtained na(Nagappan et al., 2006).

Following are the notable works in this field, although somewhat limited in number:

1. Wahyudin *et al.* (Wahyudin et al., 2008) examined the

combined effects of product and project metrics in the development of an improved predictive model. Their study used project metrics collected from Apache MyFaces project family over a span of two years. Through, correlation analysis, selected project metrics revealed a strong correlation between product metrics. To reduce the dimensionality of the combination of metrics, stepwise regression was applied. Their work shows the importance of the combination of metrics, without deliberating interaction between metrics.

2. D'Ambros *et al.* (DAmbros et al., 2012) Ambros statistically analyzed the benefits of utilizing a combination of source code metrics and other metrics derived using information theory to predict bugs. The same authors earlier showed the comparative advantages of using the combination of CK and other object-oriented metrics (DAmbros et al., 2010). They created a bug prediction data set and made it public. The same data set is being used in our research.
3. Lee *et al.* (Lee et al., 2011) proposed 56 micro interaction metrics (MIMs) capturing developer's behavioral pattern stored in Mylyn data. Metrics associated with behavioral pattern measures developer interaction with the development environment, for example, file editing, time spent on an event, etc. they build both classification and regression models using MIM in isolation and as well as in combination with other traditional metrics and empirically analyzed their effect on software quality. This experimental data of their study is freely available for future research purposes.

This combined metrics approach of fault prediction may utilize different metrics selected from within a single project or across multiple projects. Most metrics developed for process, products and people relate to one another; therefore their combination will lead to the issue of appropriate selection of candidate metrics and take their interaction effect into account.

5. Conclusion

This paper delineates metrics, metrics suite and their usage to the applied data analytic techniques. Although developments of models make use of different kinds of metrics, the review of the literature presented here essentially focuses on the Object oriented metrics. In comparison to, procedural language based system, Object Oriented (OO) technology based systems introduce new abstractions and building blocks. Therefore, development of the new set of metrics and fault prediction models will foster quality in the developed software. The advantages of combining metrics, while implementing a metrics program in an organisation needs further investigation.

References

Aggarwal, K.K., Singh, Y., Kaur, A., Malhotra, R., 2006. Empirical Study of Object-Oriented Metrics. Journal of Object Technology 5, 149–173.

Aggarwal, K.K., Singh, Y., Kaur, A., Malhotra, R., 2009. Empirical analysis for investigating the effect of object-oriented metrics on fault proneness: a replicated case study. Software Process: Improvement and Practice 14, 39–62.

Aljahdali, S., Sheta, A.F., 2011. Predicting the Reliability of Software Systems Using Fuzzy Logic, in: Information Technology: New Generations (ITNG), 2011 Eighth International Conference on. pp. 36–40.

Alshayeb, M., Li, W., 2003. An empirical validation of object-oriented metrics in two different iterative software processes. Software Engineering, IEEE Transactions on 29, 1043–1049.

Anh, N.D., 2010. The impact of design complexity on software cost and quality.

Arisholm, E., Briand, L.C., Johannessen, E.B., 2010. A systematic and comprehensive investigation of methods to build and evaluate fault prediction models. Journal of Systems and Software 83, 2–17.

Azar, D., Vybihal, J., 2011. An ant colony optimization algorithm to improve software quality prediction models: Case of class stability. Information and Software Technology 53, 388–393.

Babic, D., 2012. Adaptive Software Fault Prediction Approach Using Object-Oriented Metrics.

Back, T., Fogel, D.B., Michalewicz, Z., 1997. Handbook of evolutionary computation. IOP Publishing Ltd.

Bai, C.G., Hu, Q.P., Xie, M., Ng, S.H., 2005. Software failure prediction based on a Markov Bayesian network model. Journal of Systems and Software 74, 275–282.

Bansiya, J., Davis, C.G., 2002. A hierarchical model for object-oriented design quality assessment. Software Engineering, IEEE Transactions on 28, 4–17.

Basili, V.R., Briand, L.C., Melo, W.L., 1996. A validation of object-oriented design metrics as quality indicators. Software Engineering, IEEE Transactions on 22, 751–761.

Beecham, S., Baddoo, N., Hall, T., Robinson, H., Sharp, H., 2006. Protocol for a systematic literature review of motivation in software engineering.

Beecham, S., Baddoo, N., Hall, T., Robinson, H., Sharp, H., 2008. Motivation in Software Engineering: A syste literature review. Information and Software Technology 50, 860–878.

Bibi, S., Tsoumakas, G., Stamelos, I., Vlahavas, I., 2008. Regression via Classification applied on software defect estimation. Expert Systems with Applications 34, 2091–2101.

Biolchini, J., Mian, P.G., Natali, A.C.C., Travassos, G.H., 2005. Systematic review in software engineering. System Engineering and Computer Science Department COPPE/UFRJ, Technical Report ES 679.

Bouktif, S., Ahmed, F., Khalil, I., Antoniol, G., 2010. A novel composite model approach to improve software quality prediction. Information and Software Technology 52, 1298–1311.

Brady, A., Menzies, T., 2010. Case-based reasoning vs parametric models for software quality optimization, in: Proceedings of the 6th International Conference on Predictive Models in Software Engineering. p. 3.

Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., 1993. Classification and Regression Trees, Wadsworth International Group, Belmont, CA, 1984. There is no corresponding record for this reference 1–359.

Briand, L.C., Daly, J., Porter, V., Wust, J., 1998. A comprehensive empirical validation of design measures for object-oriented systems, in: Software Metrics Symposium, 1998. Metrics 1998. Proceedings. Fifth International. pp. 246–257.

Briand, L.C., Wust, J., Daly, J.W., Porter, V., 2000. Exploring the relationships between design measures and software quality in object-oriented systems. Journal of Systems and Software 51, 245–273.

Caglayan, B., Tosun, A., Miranskyy, A., Bener, A., Ruffolo, N., 2010. Usage of multiple prediction models based on defect categories, in: Proceedings of the 6th International Conference on Predictive Models in Software Engineering. p. 8.

Carapucca, R., Others, 1994. Candidate metrics for object-oriented software within a taxonomy framework. Journal of Systems and Software 26, 87–96.

Catal, C., 2011. Software fault prediction: A literature review and current trends. Expert Systems with Applications 38, 4626–4636.

Catal, C., Diri, B., 2007. Software fault prediction with object-oriented metrics based artificial immune recognition system.

Product-Focused Software Process Improvement 300–314.

Catal, C., Diri, B., 2009. A systematic review of software fault prediction studies. Expert Systems with Applications 36 7346–7354.

Catal, C., Sevim, U., Diri, B., 2011. Practical development of an Eclipse-based software fault prediction tool using Naive Bayes algorithm. Expert Systems with Applications 38, 2347–2353.

Chidamber, S.R., Kemerer, C.F., 1994. A metrics suite for object oriented design. Software Engineering, IEEE Transactions on 20, 476–493.

Chiu, N., 2011. Combining techniques for software quality classification: An integrated decision network approach. Expert Systems with Applications 38, 4618–4625.

Chowdhury, I., Zulkernine, M., 2011. Using complexity, coupling, and cohesion metrics as early indicators of vulnerabilities. Journal of Systems Architecture 57, 294–313.

Corazza, A., Di Martino, S., Ferrucci, F., Gravino, C., Sarro, F., Mendes, E., 2010. How effective is tabu search to configure support vector regression for effort estimation?, in: Proceedings of the 6th International Conference on Predictive Models in Software Engineering. p. 4.

Couto, C., Montandon, J.E., Silva, C., Valente, M.T., 2012. Static correspondence and correlation between field defects and warnings reported by a bug finding tool. Software Quality Journal 1–17.

DAmbros, M., Lanza, M., Robbes, R., 2010. An extensive comparison of bug prediction approaches, in: Mining Software Repositories (MSR), 2010 7th IEEE Working Conference on. pp. 31–41.

DAmbros, M., Lanza, M., Robbes, R., 2012. Evaluating defect prediction approaches: a benchmark and an extensive comparison. Empirical Software Engineering 17, 531–577.

Death, G., Fabricius, K.E., 2000. Classification and regression trees: a powerful yet simple technique for ecological data analysis. Ecology 81, 3178–3192.

Dejaeger, K., Verbraken, T., Baesens, B., 2012. Towards comprehensible software fault prediction models using Bayesian network classifiers. Software Engineering, IEEE Transactions on.

Di Martino, S., Ferrucci, F., Gravino, C., Sarro, F., 2011. A genetic algorithm to configure support vector machines for predicting fault-prone components, in: Product-Focused Software Process Improvement. Springer, pp. 247–261.

Dick, S., Meeks, A., Last, M., Bunke, H., Kandel, A., 2004. Data mining in software metrics databases. Fuzzy Sets and Systems 145, 81–110.

Dvzeroski, S., Drumm, D., 2003. Using regression tree identify the habitat preference of the sea cucumber (< i> Holothuria leucospilota</i>) on Rarotonga, Cook Islands. Ecological Modelling 170, 219–226.

Dybaa, T., Kampenes, V.B., Sjo berg, D.I.K., 2006. A systematic review of statistical power in software engineering experiments. Information and Software Technology 48, 745–755.

E Abreu, F., Melo, W., 1996. Evaluating the impact of object-oriented design on software quality, in: Software Metrics Symposium, 1996., Proceedings of the 3rd International. pp. 90–99.

Easterbrook, S., Singer, J., Storey, M.-A., Damian, D., 2008. Selecting empirical methods for software engineering research, in: Guide to Advanced Empirical Software Engineering. Springer, pp. 285–311.

Ebert, C., Dumke, R., 2007. Measurement Foundations. Software Measurement: Establish Extract Evaluate and Execute 41–72.

El Emam, K., Benlarbi, S., Goel, N., Rai, S.N., 2001. The confounding effect of class size on the validity of object-oriented metrics. Software Engineering, IEEE Transactions on 27, 630–650.

Elish, K.O., Elish, M.O., 2008. Predicting defect-prone software modules using support vector machines. Journal of Systems and Software 81, 649–660.

Fenton, N., Krause, P., Neil, M., 2002. Software measurement: Uncertainty and causal modeling. Software, IEEE 19, 116–122.

Frank, E., Wang, Y., Inglis, S., Holmes, G., Witten, I.H., 1998. Using model trees for classification. Machine Learning 32, 63–76.

Gay, G., 2010. A baseline method for search-based software engineering, in: Proceedings of the 6th International Conference on Predictive Models in Software Engineering. p. 2.

Gokhale, S.S., Lyu, M.R., 1997. Regression tree modeling for the prediction of software quality, in: Proceedings of the Third ISSAT International Conference on Reliability and Quality in Design. pp. 31–36.

Guo, L., Ma, Y., Cukic, B., Singh, H., 2004. Robust prediction of fault-proneness by random forests, in: Software Reliability Engineering, 2004. ISSRE 2004. 15th International Symposium on. pp. 417–428.

Gyimothy, T., Ferenc, R., Siket, I., 2005. Empirical validation of object-oriented metrics on open source software for fault prediction. Software Engineering, IEEE Transactions on 31, 897–910.

Hall, T., Beecham, S., Bowes, D., Gray, D., Counsell, S., 2011. A systematic review of fault prediction performance in software engineering. Software Engineering, IEEE Transactions on.

Harman, M., 2010. The relationship between search based software engineering and predictive modeling, in: Proceedings of the 6th International Conference on Predictive Models in Software Engineering. p. 1.

Harman, M., Mansouri, S.A., Zhang, Y., 2009. Search based software engineering: A comprehensive analysis and review of trends techniques and applications. Department of Computer Science, King's College London, Tech. Rep. TR-09-03.

Harman, M., McMinn, P., de Souza, J.T., Yoo, S., 2012. Search based software engineering: Techniques, taxonomy, tutorial, in: Empirical Software Engineering and Verification. Springer, pp. 1–59.

Harrison, R., Counsell, S.J., Nithi, R.V., 1998. An evaluation of the MOOD set of object-oriented software metrics. Software Engineering, IEEE Transactions on 24, 491–496.

Hitz, M., Montazeri, B., 1995. Measuring coupling and cohesion in object-oriented systems, in: Proceedings of the International Symposium on Applied Corporate Computing. pp. 75–76.

Hong, Y., Kim, W., Joo, J., 2010. Prediction of defect distribution based on project characteristics for proactive project management, in: Proceedings of the 6th International Conference on Predictive Models in Software Engineering. p. 15.

Janes, A., Scotto, M., Pedrycz, W., Russo, B., Stefanovic, M., Succi, G., 2006. Identification of defect-prone classes in telecommunication software systems using design metrics. Information Sciences 176, 3711–3734.

Jones, T.C., 2008. Applied Software Measurement: Global Analysis of Productivity and Quality, 3E.

Kaner, C., Bond, W.P., 2004. Software engineering metrics: What do they measure and how do we know? methodology 8, 6.

Kanmani, S., Uthariaraj, V.R., Sankaranarayanan, V., Thambidurai, P., 2007. Object-oriented software fault prediction using neural networks. Information and Software Technology 49, 483–492.

Khoshgoftaar, T.M., Allen, E.B., Deng, J., 2002. Using regression trees to classify fault-prone software modules. Reliability, IEEE Transactions on 51, 455–462.

Khoshgoftaar, T.M., Allen, E.B., Hudepohl, J.P., Aud, S.J., 1997. Application of neural networks to software quality modeling of a very large telecommuni system. Neural Networks, IEEE Transactions on 8, 902–909.

Khoshgoftaar, T.M., Seliya, N., 2003. Fault prediction modeling for software quality estimation: Comparing commonly used techniques. Empirical Software Engineering 8, 255–283.

Khoshgoftaar, T.M., Seliya, N., Sundaresh, N., 2006. An empirical study of predicting software faults with case-based reasoning. Software Quality Journal 14, 85–111.

Kitchenham, B., Brereton, P., Budgen, D., Turner, M., Bailey, J., Linkman, S., 2009. Systematic literature reviews in software engineering-a systematic literature review. Information and software technology 51, 7–15.

Kitchenham, B.A., Pfleeger, S.L., Pickard, L.M., Jones, P.W., Hoaglin, D.C., El Emam, K., Rosenberg, J., 2002. Preliminary guidelines for empirical research in software engineering. Software Engineering, IEEE Transactions on 28, 721–734.

Koru, A.G., Liu, H., 2005. Building effective defect-prediction models in practice. Software, IEEE 22, 23–29.

Laird, L.M., Brennan, M.C., 2006. Software measurement and estimation: a practical approach. John Wiley and Sons.

Lanubile, F., Visaggio, G., 1997. Evaluating predictive quality models derived from software measures: lessons learned. Journal of Systems and Software 38, 225–234.

Lavazza, L., Robiolo, G., 2010. The role of the measure of functional complexity in effort estimation, in: Proceedings of the 6th International Conference on Predictive Models in Software Engineering. p. 6.

Lee, T., Nam, J., Han, D., Kim, S., In, H.P., 2011. Micro interaction metrics for defect prediction., in: SIGSOFT FSE. pp. 311–321.

Lessmann, S., Baesens, B., Mues, C., Pietsch, S., 2008. Benchmarking classification models for software defect prediction: A proposed framework and novel findings. Software Engineering, IEEE Transactions on 34, 485–496.

Li, W., Henry, S., 1993. Object-oriented metrics that predict maintainability. Journal of systems and software 23, 111–122.

Li, W., Henry, S., Selig, C., 1991. Measuring Ada Design to Predict Maintainability, in: 9th Annual National Conference on Ada Technology. pp. 107–113.

Lorenz, M., Kidd, J., 1994. Object-oriented software metrics: a practical guide. Prentice-Hall, Inc.

Luo, Y., Ben, K., Mi, L., 2010. Software metrics reduction for fault-proneness prediction of software modules, in: Network and Parallel Computing. Springer, pp. 432–441.

Mair, C., Shepperd, M., 2011. Human judgement and software metrics: vision for the future, in: Proceedings of the 2nd International Workshop on Emerging Trends in Software Metrics. pp. 81–84.

Malhotra, R., Kaur, A., Singh, Y., 2010. Empirical validation of object-oriented metrics for predicting fault proneness at different severity levels using support vector machines. International Journal of System Assurance Engineering and Management 1, 269–281.

Marcus, A., Poshyvanyk, D., Ferenc, R., 2008. Using the conceptual cohesion of classes for fault prediction in object-oriented systems. Software Engineering, IEEE Transactions on 34, 287–300.

Matsumoto, S., Kamei, Y., Monden, A., Matsumoto, K., Nakamura, M., 2010. An analysis of developer metrics for fault prediction, in: Proceedings of the 6th International Conference on Predictive Models in Software Engineering. p. 18.

Maxwell, K., 2002. Applied statistics for software managers. Prentice Hall.

McCabe, T.J., 1976. A complexity measure. Software Engineering, IEEE Transactions on 1, 308–320.

Mende, T., 2010. Replication of defect prediction studies: problems, pitfalls and recommendations, in: Proceedings of the 6th International Conference on Predictive Models in Software Engineering. p. 5.

Menzies, T., Milton, Z., Turhan, B., Cukic, B., Jiang, Y., Bener, A., 2010. Defect prediction from static code features: current results, limitations, new approaches. Automated Software Engineering 17, 375–407.

Menzies, T., Shepperd, M., 2012. Special issue on repeatable results in software engineering prediction. Empirical Software Engineering 17, 1–17.

Mertik, M., Lenic, M., Stiglic, G., Kokol, P., 2006. Estimating software quality with advanced data mining techniques, in: Software Engineering Advances, International Conference on. p. 19.

Meziane, F., Vadera, S., 2010. Artificial intelligence applications for improved software engineering development: new prospects. Information Science Reference.

Michura, J., Capretz, M.A.M., Wang, S., 2013. Extension of Object-Oriented Metrics Suite for Software Maintenance. ISRN Software Engineering 2013.

Misra, S., 2011. Evaluation Criteria for Object-oriented Metrics. Acta Polytechnica Hungarica 8, 110–136.

Misra, S., Adewumi, A., 2014. Object-Oriented Cog Complexity Measures: An Analysis. Handbook of Research on Innovations in Systems and Software Engineering 150.

Misra, S., Koyuncu, M., Crasso, M., Mateos, C., Zunino, A., 2012. A suite of cognitive complexity metrics, in: Computational Science and Its Applications-ICCSA 2012. Springer, pp. 234–247.

Muzaffar, Z., Ahmed, M.A., 2010. Software development effort prediction: A study on the factors impacting the accuracy of fuzzy logic systems. Information and Software Technology 52, 92–109.

Nagappan, N., Ball, T., Zeller, A., 2006. Mining metrics to predict component failures, in: Proceedings of the 28th International Conference on Software Engineering. pp. 452–461.

Pai, G.J., Dugan, J.B., 2007. Empirical analysis of software fault content and fault proneness using Bayesian methods. Software Engineering, IEEE Transactions on 33, 675–686.

Pandey, A.K., Goyal, N.K., 2009. A Fuzzy Model for Early Software Fault Prediction Using Process Maturity and Software Metrics. International Journal of Electronics Engineering 1, 239–245.

Perry, D.E., Porter, A.A., Votta, L.G., 2000. Empirical studies of software engineering: a roadmap, in: Proceedings of the Conference on The Future of Software Engineering. pp. 345–355.

Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M., 2008. Systematic mapping studies in software engineering, in: 12th International Conference on Evaluation and Assessment in Software Engineering. p. 1.

Quinlan, J.R., 1992. Learning with continuous classes, in: Proceedings of the 5th Australian Joint Conference on Artificial Intelligence. pp. 343–348.

Radjenovic, D., Herico, M., Torkar, R., Zivkovic, A., 2013. Software fault prediction metrics: A systematic literature review. Information and Software Technology 55, 1397–1418.

Raffo, D., Harrison, W., Vandeville, J., 2000. Coordinating models and metrics to manage software projects. Software Process: Improvement and Practice 5, 159–168.

Raj Kiran, N., Ravi, V., 2008. Software reliability prediction by soft computing techniques. Journal of Systems and Software 81, 576–583.

Rodriguez, D., Herraiz, I., Harrison, R., 2012. On software engineering repositories and their open problems, in: Realizing Artificial Intelligence Synergies in Software Engineering (RAISE), 2012 First International Workshop on. pp. 52–56.

Runeson, P., Andersson, C., Thelin, T., Andrews, A., Berling, T., 2006. What do we know about defect detection methods?[software testing]. Software, IEEE 23, 82–90.

Runkler, T.A., 2012. Data Analytics: Models and Algorithms for Intelligent Data Analysis. Vieweg Teubner Verlag.

Seaman, C.B., 1999. Qualitative methods in empirical studies of software engineering. Software Engineering, IEEE Transactions on 25, 557–572.

Shepard, T., Lamb, M., Kelly, D., 2001. More testing should be taught. Communications of the ACM 44, 103–108.

Singer, J., Vinson, N., 2002. Ethical issues in empirical studies of software engineering.

So, S.S., Cha, S.D., Kwon, Y.R., 2002. Empirical evaluation of a fuzzy logic-based software quality prediction model. Fuzzy Sets and Systems 127, 199–208.

Succi, G., Pedrycz, W., Stefanovic, M., Miller, J., 2003. Practical assessment of the models for identification of defect-prone classes in object-oriented commercial systems using design metrics. Journal of Systems and Software 65, 1–12.

Tang, M.-H., Kao, M.-H., Chen, M.-H., 1999. An empirical study on object-oriented metrics, in: Software Metrics Symposium, 1999. Proceedings. Sixth International. pp. 242–249.

Tegarden, D.P., Sheetz, S.D., Monarchi, D.E., 1995. A software complexity model of object-oriented systems. Decision Support Systems 13, 241–262.

Thwin, M.M.T., Quah, T.-S., 2003. Application of neural networks for software quality prediction using object-oriented metrics, in: Journal of Systems and Software. Elsevier, pp. 147–156.

Tichy, W.F., 1998. Should computer scientists experiment more? Computer 31, 32–40.

Uysal, I., Guvenir, H.A., 1999. An overview of regression techniques for knowledge discovery. Knowledge Engineering Review 14, 319–340.

Vandecruys, O., Martens, D., Baesens, B., Mues, C., De Backer, M., Haesen, R., 2008. Mining software repositories for comprehensible software fault prediction models. Journal of Systems and software 81, 823–839.

Verma, H.K., Sharma, V., 2010. Handling imprecision in inputs using fuzzy logic to predict effort in software development, in: Advance Computing Conference (IACC), 2010 IEEE 2nd International. pp. 436–442.

Wagner, S., 2013. Quality Planning, in: Software Product Quality Control. Springer, pp. 91–110.

Wahyudin, D., Schatten, A., Winkler, D., Tjoa, A.M., Biffl, S., 2008. Defect Prediction using Combined Product and Project Metrics-A Case Study from the Open Source, in: Software Engineering and Advanced Applications, 2008. SEAA’08. 34th Euromicro Conference. pp. 207–215.

Wang, Y., Shao, J., 2003. Measurement of the cognitive functional complexity of software, in: Cognitive Informatics, 2003. Proceedings. The Second IEEE International Conference on. pp. 67–74.

Witten, I.H., Frank, E., 2005. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.

Xing, F., Guo, P., Lyu, M.R., 2005. A novel method for early software quality prediction based on support vector machine, in: Software Reliability Engineering, 2005. ISSRE 2005. 16th IEEE International Symposium on. p. 10–pp.

Xu, J., Ho, D., Capretz, L.F., 2008. An empirical validation of object-oriented design metrics for fault prediction. Journal of Computer Science 4, 571.

Yan, X., Su, X.G., 2009. Linear regression analysis: theory and computing. World Scientific Publishing Company.

Yang, B., Yao, L., Huang, H.-Z., 2007. Early software quality prediction based on a fuzzy neural network model, in: Natural Computation, 2007. ICNC 2007. Third International Conference on. pp. 760–764.

Yuhas, B., Ansari, N., 2012. Neural networks in telecommunications. Springer Publishing Company, Incorporated.

Zheng, J., 2010. Cost-sensitive boosting neural networks for software defect prediction. Expert Systems with Applications 37, 4537–4543.

An Architectural Framework for E-Voting Administration

Shadreck Mudziwepasi

&

Mfundo Shakes Scott

University of Fort Hare, Department of Computer Science, P/Bag X1314, Alice 5700, RSA
{smudziwepasi, sscott}@ufh.ac.za

Abstract- One of the key areas of concentration in achieving harmonious democracy is transparency in the electoral processes. Some countries on the African continent such as Ghana and Kenya have recently had issues of doubt and mistrust of the administration and the management of their Electoral Commission and hence a suspicion of election fraud which has prone threats of violence, economic declination and on the peak, legal implications. There was a claim of double registration, duplicated ballots, lost ballots, wrong count of ballots, failure of biometric registration system, impersonation, and alteration of counted votes in the immediate past election in countries such as Ghana, which led to series of court cases. E- Voting brings about a suitable solution to these. Available Literature at present exclusively reveals that most e-voting systems have presented several failures in design. This raises eyebrows concerning the technical and procedural controls on whether they are sufficient to guarantee trustworthy voting. The best methods possible should be applied in order to come up with the best solutions based on a framework that thoroughly addresses the requirements and standards. Therefore, this paper seeks to optimize the voting processes and governance of the Electoral Commission of respective countries by proposing a trustable e-voting theoretical framework which dwells on biometric data of various candidates as the basis for encryption of ballot, dedicated channel for transmission of counted ballots and/or connecting and disconnecting the database server before and after voting. Various literatures are considered to help propose a robust framework

Keywords: E-Diary Services, UFH network, framework, Development

1. Introduction

The Elections and voting practices are of very much importance to all countries that practice democracy. It is through the process of elections for which citizens have the opportunity to choose the leaders and representatives of their choice (Rexa et al., 2012) At present in most countries, the fundamental right of choosing a leader using a voting system is done mainly in manual and paper form (Kingsley et al., 2014). However, the administration of the voting process when done manually makes it prone to various electoral problems such as over-voting, wrong count, impersonation, lost ballots, spoilt ballot, declining turnout of voters, difficulty of auditing after voting and poor documentation (Nu'man et al., 2012). This situation calls for immediate attention to the methods used in voting. Around Africa, some countries such as Sierra Leon,

Rwanda and Uganda have had riots because of the poor administration of the electoral process and in all of these countries, the manual paper forms are used for voting (Bamiah et al., 2010). A robust framework to implement E-Voting would really prove to be vital in addressing these challenges

At present and in most countries, the norm is that an Electoral commission (EC) is mandated to be in charge of free and fair elections (Alkasar et al., 2014). The EC is responsible for the organization of elections in most countries (Bamiah et al., 2010).

However, election administration is bringing about a new dimension in the history of most countries, especially where the major opposing parties have doubts of the results and hence they launch court cases against the winning party. The parties would disagree with the results from the EC and in most cases claim that they are fraudulent. This calls for the fact that an effective and trustworthy system would be required to replace the manual system to enhance the trust of the citizens in the voting system (Alkasar et al., 2014).

Therefore, this research work seeks to examine the lapses in the existing voting system and propose a trustable e-voting system framework for which when adopted and implemented would solve the majority of the problems faced. Besides system analysis against requirements, it is also important to carry out an amalicious circumstances scruitiny on occasions when the execution model is susceptible to attack. We therefore outspread system specification by guarding it against compromising attacks (Yeboah et al., 2013). This was supportive in our aim to spot mislaid requirements and conventions respectively regarding system specification. Also, this allowed us to outline counter measure initiatives that can be used alternatively when the system malfunctions to increase the reliability of the system throughout its design.

It is acknowledged in this work that the central problem we find in e-voting applications specification and verification is the challenges in modeling attacks since the different types of attack relay across the structure of the unique performance models, resulting in difficulties in incremental verification (Patey et al., 2014). A robust framework would therefore be necessary to implement a useful e-voting system.

2. Related Research

E-voting is advantageous because of it ensures enhanced turn out and easy accessibility especially for disabled and/or or impaired people including improved efficiency and reliability.

However, e-voting adoption in various countries has been poor and slow and/or being the cause of debate and controversy. This is largely as a result of the largely poor implementation of (some of) the prototype systems currently deployed for elections in other countries such as the United States of America (USA), according to literature (Heitmeyer et al., 2008).

Present literature also shows that such systems have major and serious flaws in specifications and design. Thus, such weaknesses expose the system, and consequently make elections vulnerable to malfunctioning and various threats and attacks, ranging from a denial of service to result alteration. There are several research works done by some researchers in e-voting security and trust issues.

(Alkasar et al., 2014), proposes a framework to manage a secure trustworthy E-voting system, by securing each and every side of the system from its initial stage to finishing stage through implementation of the Trusted Platform Technology (TPM). The TPM serve as a chain of trust that combines hardware and software to provide trusted client device.

However, in their research they failed to provide the design of the TPM and how it was used to secure the vote, channel, the computers, and mobile phones in their framework. Also, on their proposed framework the entire voting process is obscured from the voter and polling agents. They only get to know the result from the polling station only when the entire voting process has ended. This will negatively affect the trust of the voters.

In (Yeboah et al., 2013) is a proposed e-voting framework that will enhance the security of the immediate manual system if adopted. To enhance the security, they implemented it using smart cards and digital certificates. However their framework is expensive because at every polling station they implemented 2 ARC (Archive) redundant servers which invariable stored small amount of records. Also, to secure records, the systems were configured by the national election commission. Their research did not cover the polling agents at this level, for which it can implicate the trust of the system by the voters.

In (Patey et al., 2011), a research work proposed a framework which was aimed at improving authentication and transparency in e-voting systems.

Their systems framework was to replace the manual system so then their citizens will be able to vote from any polling station. This concept was derived from queue listing dynamic technique which is centered on arrivals and identification of the subsequent voters at the polling station. However, the system could not address how the centralized database could be protected to check for content; whether there are votes or no votes already in the in database before voting starts. Again, their research work did not consider the integrity of casted votes during the time of voting.

The work of Alkassar in (Jones et al., 2009) proposed a solution to security of online voting systems bringing to bare how unsecured malware and corrupt voters activities on the voting system could affect the trustworthiness of the voting process and the voting system entirely. Their solution was based on Trusted Computing in combination with secure operating systems. However, they did not consider the security breach based on amount of time spent within the network and the number of attempts of logging onto the system.

Their framework could have been very much effective if a defined set of parameters were identified to avoid breaches to use of the voting system. Their framework could not detect exactly who is voting, this is because it is done online. The framework lacks physical system administration and monitoring. A system's reliable specification behavior is achieved only if the best techniques are employed.

In this regards, a number of technical approaches to address (some of) the issues mentioned above have been devised and are hereby outlined in this research work. Among these include the implementation of formal methods and robust frameworks which have been proven to improve the reliability and efficiency of complicated systems. No mechanisms were defined to tell genuine citizens are identified to vote anywhere.

Therefore, there is the need for a framework that identifies each voter before the ballot is casted.

3. Framework for the Existing Manual System

The existing frame works exemplified by those mentioned in the related research section can be illustrated by Figure 1 below. This figure basically shows the frame work for the existing voting system currently adopted for use in some countries across the globe (Juels et al., 2005).

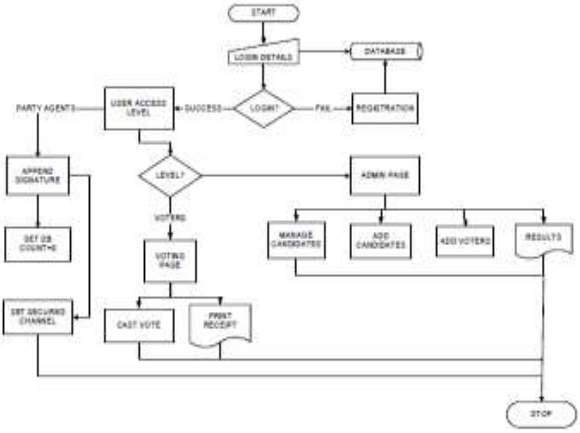


Figure 1: Flowchart of the Existing Voting System

3.1 Current Voting Process

Without any unforeseen circumstances, voting takes place only on the day of election normally starting from 7am up until 7pm. This is not to say one cannot vote when it is 7pm while he/she is already in queue to vote (Alkasar et al., 2014). It is only people who get to the polling stations at 7pm or after that would not be allowed to vote. The voter goes through the following processes:

- The voter is required to check His/her name in the reference list so as to identify himself/herself as an eligible voter. This is done by showing your voters' ID card to the officials who would in turn cross check to see whether you are in the voters' register and also place your thumb on the biometric machine to verify if you are truly the card bearer (Kremer et al., 2014).
- After all necessary information has been checked out, your finger would be dipped into an indelible ink. This is also a measure to prevent double voting. But one must be sure not to stain the ballot paper with that finger since a soiled ballot paper would be rejected (Lowry et al., 2014).
- After receiving your ballot paper, one should carefully check whether it has the ECs official stamp on it before he/she goes to vote otherwise that ballot paper is considered invalid .
- Voting is done in mainly three ways; presidential, parliamentary and local government. After receiving the presidential ballot first, you would proceed to a voting booth where you find an ink to dip your thumb in and then carefully vote in the space provided for your candidate of choice. After which you wipe your finger first and gently fold the

paper into the ballot boxes.

- You then proceed to the next table for the parliamentary ballot paper and also follow the same procedure as the presidential one above and drop the ballot paper in the parliamentary ballot box after which you will also do the same for the local government ballot.

3.2 Predominant Challenges of the Manual System Poor

- Documentation and Recording: In the past elections in countries such as Ghana, there were several situations of poor recording of total ballots in some of the polling stations. For example; 270 being written in words as twenty seven zero. Their respective meanings are completely different.
- Alteration of Votes: Votes can easily be manipulated because they are directly recorded on paper. The records can be exposed to any voter or official with malicious intention. According to (Martinelli et al., 2002), electoral personnel always replicate the votes which at the normal circumstance would not have been so as compared to e- voting which is claimed to be devoid of such.
- Voter Error: Voters sometimes makes errors. For example a voter may unintentionally thumb print against the picture of a candidate for which he did not intend to have voted for. However, you cannot make changes to the selected option. Also, if a voter does not fold their ballot papers well, the ink can spread to another candidate column, and hence the vote is disqualified and nullified. The Official website of Electoral commission, R&M Department of South Africa provides a summary of rejected ballots from 1992 to 2008 as follows: 3.03, 1.53, 1.58, and 2.13 2.32. by percentage through observation. It is clear that from year to year, the total percentage of spoilt ballots is increasing. In a 2012, December election in Ghana, 251 720 out of 11,246,982 total votes casted were nullified because either voters voted for two different parties at the same time or they were left blank.
- Deferrals in Showcasing Final Results: According to (McDaniel et al., 2014), it has been ascertained that it can take up to 3 days for the IEC of South Africa to eventually publish election results worse still a national election. This situation

prevails because they manually do the collation of results from all the various polling stations, constituency and then the national levels, which is very tedious and cumbersome to be finished within a short period of time such as an hour.

- Ballot Design and Count: Recently, biometric registration and verification were introduced into the electoral processes. However, when voting is done, all the ballot papers are mixed and no voter can be linked to any ballot paper. Also, counting is very difficult, because ballot papers are mixed up in one ballot box and are unorganized. After voting, counting is manually done for each party. Errors may occur during a large count and hence would affect the result.
- Unsecured Medium for Transfer of Ballot Count: Transferring votes from one polling station to the constituency involved can result in inconsistencies and it is was alleged that for most voting nations, there is always a suspicion of transferred votes not matching counted votes at a polling station (Kolano et al., 2007). Methods for transfer would therefore have to be improved to close up on any loopholes.

4. Proposed E-Voting System

4.1 System Requirements

Before system design, a comprehensive requirements gathering process is necessary. These requirements include generic, system and election-specific requirements as presented in Figure 2 below.

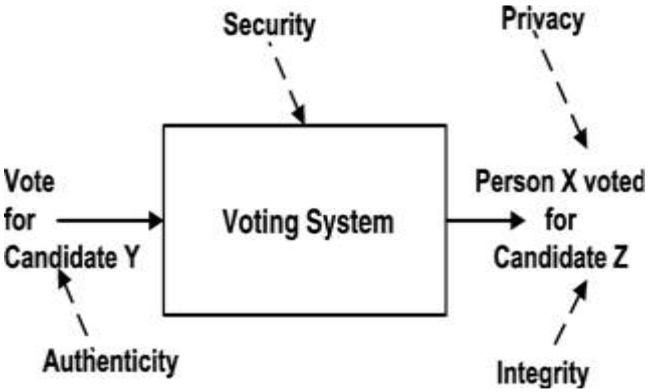


Figure 2: Set of Requirements

4.1.1 System Requirements

The generic requirements apply to any voting system (Patey et al., 2014). These requirements, as shown in Figure.1 above and are further listed and briefly explained below:

- Authenticity: only those eligible registered voters are able to vote;
- Integrity/accuracy: A vote cannot be altered in any way once casted. Valid votes count and invalid votes become spoilt;
- Privacy: A voters vote is his/her own secret;
- Security: No one can temper with a vote throughout the voting process;
- Democracy: qualified voters must vote only once.
- Confidentiality: The system should be very confidential and information should be kept as thus

4.1.2 System-specific requirements

All the system-requirements in general refer to all those requirements that have a lot to do specifically with the on-line digital electronic system of voting.

The system-specific requirements include:

- Multi-user: numerous voters can vote at once;
- Multi-elections: numerous elections can be run at once;
- Accessibility: The accessibility of the system by voters is generally in reference to the ability of the voters to be able to cast their ballots any time on www using any device;
- Availability: this highly speaks volume in the ability of the system to be open to anyone and available during the time of the elections and also during the period of election campaigns.

4.1.3 Election-Specific Requirements

As stated earlier, these types of requirements highly refer to those needs that are needed in any particular singled out election (Alkasar et al., 2014). If we can as an example consider the election specific requirements for student council election:

- A voter is supposed to be a full time registered student of the university;
- A candidate is supposed to be a full time registered student of the university as well;
- A candidate is supposed to have completed have completed at least 2 semester blocks at the university and maybe a particular GPA that can be set is supposed to have been obtained;
- A candidate must have a 1 year term limit.
- A candidate can also vote.

4.2 E-voting System Organization

A architectural system for voting

online and its administration is given in Figure 3 below and as shown, it is supposed to consists of a several components and each of those is explained here.

4.2.1 The Subsequent Election Database monitoring and System of Database Administration

This stores the information and all associated features of the elections in digital form, this will include all information about the candidates and even the voters roll as well as information regarding the polling stations and the officers including the locations of the stations and the subsequent voting times etc. An example of the technology used for such systems include Oracle.

4.2.2 The Web Server and the Web Pages

The main functionality of this feature (server) is to connect our system to IP. Furthermore, it keeps overally keeps track of the pages and the technology and functionality that is required. The web pages can be defined as either stationary/static and/or dynamic. Static web pages remain with their original data while dynamic content can be altered and modified over its entire life. Some technologies that are used in the creation of dynamic webpages include Java Server Pages (JSP).

4.2.3 The SMS Server

The Short Message Service (SMS) server is able to communicate with voters through SMS messaging. The SMS server utilizes the Global System for Mobile Communications (GSM) to relay SMS messages to all voters through a SMS network service provider. Upon completion of the entire voting campaign, the SMS server will then send all registered mobile voters a message with the election results.

It is also vital to be aware of the different e-voting system components. These will determine the budget that is supposed to be set aside for the project and they overally makes it easier to do your system requirements and gathering implementation. Furthermore, the technologies and operating systems used for these components need to be set with the latest versions to avoid any unnecessary loss of data or creation of loopholes for hackers to exploit. The components overally connects the down and uplink and are illustrated in depth in Figure 3 below.

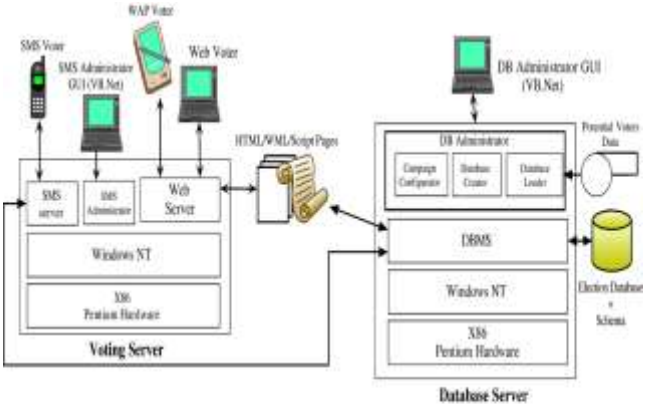


Figure 3: The Components of the Implemented E-Voting System.

4.2.4 The Access Devices

It is clear that numerous mobile devices are used to get hold of the system. These include different brands of personal computers and also different brands and types of mobile phones. The system need to define all the necessary technologies and all the protocols that will ensure compatibility of our system with all these devices. An illustration of the technologies and protocols that are available for all of these different types of devices are hereby illustrated in Figure 4 below:

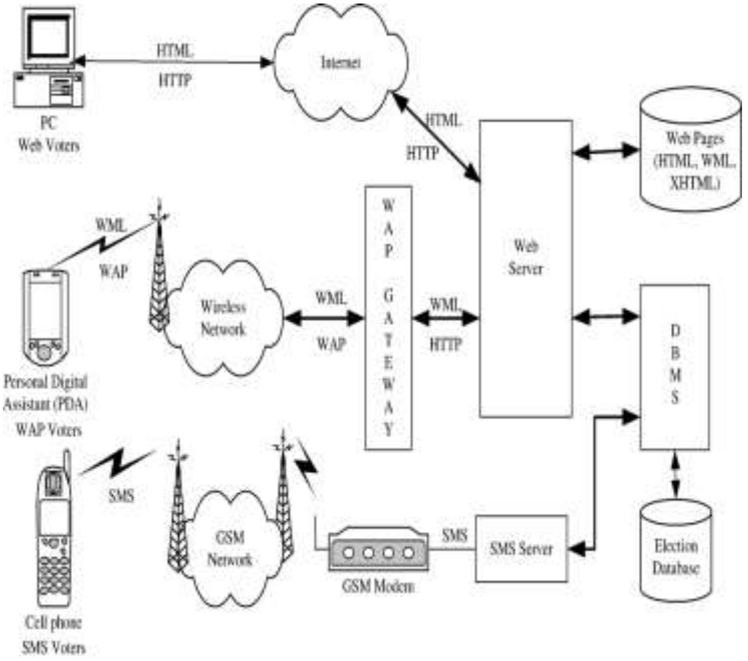


Figure 4: The Organization of a System for Electronic Voting

- **Personal/laptop Computers**: There are two types of connections that are established to and from the system. These are the wired and the wireless connectivity scheme. Computers can generally connect to the system using both systems alconnect through the wired and wireless Internet. They have a great memory for resources and high resolution screen display. Web browsers such as the Internet Explorer or Chrome or Mozilla firefox. These display a whole lot of data and information types e.g. audio and voice. This display is specified by a powerful GUI (Graphical User Interface) and languages such as the latest versions of HTML and/ or Extensible HyperText Markup Language (XHTML). These are all referred to as scripting languages and they are all also enhanced by client scripting languages that include JavaScript and VBScript. HTTP is highly utilized for the information exchange purposes.
- **WAP-enabled Handheld Devices**: These devices overally connect system using mainly a wireless connectivity scheme. It is in this category that we place mobile phones, tablets, all connected through a wireless network. They thus sometimes experience a limited network's bandwidth so they thus make full use of the Wireless Application Protocol (WAP) (Alkasar et al., 2014) framework and the closest gateway for server connection. The use of the gateway is simply to interpret and translate WAP into HTTP and the other way round. It is also very important to note that these devices are overally characterized by lesser memory sizes and processing power which results in them utilizing a limited version of the standard Internet browsers available, which in most cases would be the WAP micro-browser. This is a standard browser that can make use of the Wireless Markup Language (WML) as a specification for all its various user interfaces. The earlier versions of WML were largely based on the Extensible Markup Language (XML) and thus did not really show the major characteristics of the classical HTML language.

However, later versions of WML began to be based on the Extensible HTML (XHTML) markup language. Actually, WML now become sort of a small subset of XHTML which however requires less processing power making it most suitable for mobile devices. We can also note that WML 2.0 and beyond can support tables and simple scripting through WMLScript.

- **Regular Mobile Phones**: On the usual, these are the normal low cost, low processor power and low functionality regular cell phones. They overally utilize the SMS technology in order to establish a connection for users across system. The SMS tool is a common communication toll used by many due to low cost and user friendliness. The low cost is basically as a result of the two major influencing factors which are low cost of sending and/or receiving a message and the low cost of purchasing the mobile phone that can be able to be used in supporting this service.

  The only functionality needed in these devices is just a simple textual editor that can be utilized for composing and displaying the message. We can note one very detrimental disadvantage of an SMS which is that you sometimes cannot be able to construct a very highly interactive dialog between the mobile device and the system using a simple SMS application just like it is difficult to send some complex files with g-mail and you will need google drive to send these.

5. Proposed Architecture

Figure 5 below shows the proposed architectural framework for the E-Voting Application.

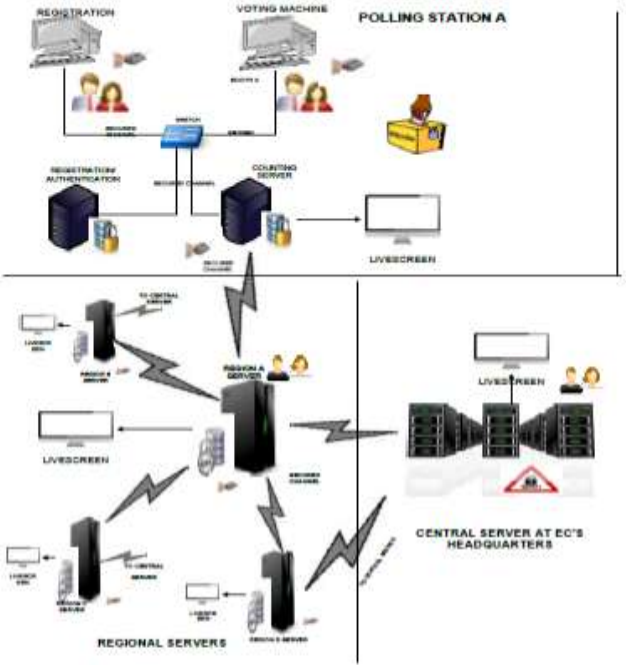


Figure 5: Proposed Framework for Implementing E-Voting Application

The architecture is based on 2-tier architecture. In this architecture, the client of the e-voting system handles the e-voting application while there is a server that will handle the database at the backend. Upon invokement by the client, it establishes a subsequent connection to the server as needed and interacts with the server. The client usually cannot see the database directly and can only access the information/data when active. Therefore server data is more secure. The client-server solution also allows many users database access simultaneously.

6. Ensuring Security

In this section, appropriate methods and techniques for ensuring security on the e-voting architecture is discussed. The network architecture for which the voting software is going to run and the design of the database are considered.

6.1. Master Database

Before the process of voting begins, party agents who are assigned cryptographic keys have to append to attest the fact that the database count is zero. The same agents would have to append their keys to encrypt and isolate the database. Then they would append to encrypt a secured channel from the system to the database server. Voters can then be allowed to cast their votes after channel to the database has been secured by the various political party representatives.

6.2. System Algorithm

* Setting database record count to zero: Use biometrics from party agents

E.g. Xi ⟶ i], where X is party

agents, R is the biometric reset sequence generated by system for

X1 → [R1]

- X2= [R1]+[R2]
- Xn = [R1]+[R2}+……[Rn]
- X1 + X2 +… Xn = empty db
  db = 0

- **Disconnecting Database from Application Before Voting**: Party agents confirm and disconnect db for authorized connections only.

E.g Xi [Ri], where X is party agents, R is the biometric reset sequence generated by system

X1 → [R1]

X2= [R1]+[R2]

Xn = [R1]+[R2}+……[Rn]

- X_1 + X_2 +… Xn disconnected db

- **Secure direct encrypted connection for eligible voters**: E.g. assuming we have polling booth [A, B, C] at a polling station,
- **At booth A, B and C**: n number of party agents is required to establish a secured connection from the system to the db.
- Booth A = X1 + X2 +… Xn
  BA = X1 + X2 +… Xn
  [BA X1 + X2 +… Xn]
- Booth B = X1 + X2 +… Xn
  BB = X1 + X2 +… Xn
  [BB X1 + X2 +… Xn]
- Booth C = X1 + X2 +… Xn
  BC = X1 + X2 +… Xn
  [BC X1 + X2 +… Xn]

7. Increased Trustworthiness

The ultimate fairness and subsequent security of all these electronic elections would largely depend on a careful requirements allocation procedures during and after holding of elections

7.1 Preliminary Level

* Auditing of the Voting System: To enhance trustworthiness of the voting system, the system would have to be tested thoroughly. The modules or scripts of the voting software and the dedicated channel would have to be also tested and tried by an auditing team made up of computer programmers and ethical hackers from each of the represented political parties. This will help the parties to understand and accept that the system is robust and hence no errors are going to emanate from the use of the system

* Registration: The registration is done at the second phase after the system has been tested thoroughly. All legible voters will come along with their voters ID card and their details as specified by the Electoral Commission would be captured including their finger prints. Upon completion of the registration, the voter is assigned a unique voters ID that is generated from a set of random

numbers.

* Securing Database and Dedicated Channel: This process is also carried out before the Election day where various representatives from the political parties are assigned biometric keys. These should be kept safe and secure by the responsible parties involved. These keys are to be appended to connect and disconnect the database before and after the main voting. This is to set the database to zero vote count and also the secure the channel for voting. The database security and protection is a very important aspect of the system. If the information in the database is in any way tempered with, then it would mean the whole system's credibility is compromised and its rating drops to unprecedented levels

7.2 Voting Level

* Voting interface: At this point, the voter is required to enter his/her unique ID and biometric data before accessing the voting panel where he/she is presented with the three main voting categories of presidential, parliamentary and local government to vote. Upon selection of each category, the voter is presented with the various candidates' names and pictures for voting. Voting for a candidate in this category is acknowledged and that category immediately disabled to prevent double voting. A receipt is then issued out to the voter which states the political party voted for. This receipt is then placed in a physical ballot box for recounting later in case of disputes. Voting process is made possible when party representatives append their signatures to secure voting channels to database. All these processes must appear graphically on a live screen.

7.3 Events after Voting

* Counting and Tallying: The subsequent ballots that were cast are shown on the live screen prior to this stage to show live results clearly to the public and to also make sure that ballots cast are equal or less than number of people registered. In case there still some doubts, the receipts placed into the ballot boxes can be recounted. Election results from each polling station are then sent to a regional server in a secured encrypted manner which is also authorized by the party agents. At the regional level, various party reps can be assigned with biometric keys to receive data from polling stations and then also send them securely to other regional servers as well as the Electoral Commission's central server. The process of exchanging data between the regional servers before sending them to the central server is to prevent the process of someone hacking into the network to change data on one

communication channel. If such a person even succeeds on that channel, we would have nine additional regional servers to cross check data for accuracy.

8. Conclusion and Future Work

Electronic voting is a system that largely concerns the behavior of all the participating individuals and the associated voting system components. The uttermost assurance of all electronic elections would definitely mean that a thorough and extensive review and investigation of all these various aspects, initiatives and associated techniques in an inclusive and integrated way. A full-bodied design would really mean our system is easy to use and will that the system continues to function even in situations when one or more components are compromised. The researchers in this paper have clearly demonstrated to a large extent considerably the use of a more authenticated and simpler user friendly approach which can be used by a voter with no difficulty when provided with the necessary training. The proposed framework when implemented will do away with most of the inconsistencies in the vote processes and will be very effective. It should be noted that however, in some African states that held elections recently such as Ghana, some voters could not be verified although their biometric data was previously collected during the registration period. Therefore an alternative to biometric data would be very important for further research and future work. Also, most of the existing e-voting systems do not consider the illiterates. Therefore an alternative scheme such as voice instruction should be further studied to see how they could be integrated into voting systems as an addition to the use of the usual input devices (such as keyboard and the mouse).

9. Acknowledgments

This is work is that has been undertaken within the TELKOM Coe in ICTD at the University of Fort Hare computer science department. It has received support in part from Telkom SA. The findings expressed here are those of the authors and none of the above sponsors accepts liability whatsoever in this regard.

10. References

Rexha, V. Neziri, and R. Dervishi. “Marching in the Direction of Framework For the Administration of E-voting:” The Case Of Ghana”. Available in International Journal of Computers and Communications. Issue 1, Volume 6, 2012

Kingsley J, K Sarpong, “Towards Improving authentication and due transparency of e-Voting Systems in the Kosovo Case”, IJCSI International Journal of Computer Science Issues, Vol. 11, Issue 3, No 2, May 2014

Rexha. B, R. Dervishi, and V.

Neziri. "2011An architectural Framework for the Adoption of E-Voting in Jordan", Retrieved from https://www.google.com.gh/#bav=on 7th August, 2013.

Nu'man.B "We are Increasing the Trustworthiness of e-Voting Systems Using Smart Cards and Digital Certificates A test case of the Kosovo Case.". Electronic Journal of e-Government Volume 10 Issue 2 2012, pp133 – 146, 2012.

Bamiah. M, A. Dehghantanha, and B. Archibald. "Towards Trustworthy Architectural voting systems" http://www.academia.edu/643254/A_Trustable_Electronic Government_Voting_ on 15th August, 2010.

Alkassar. A , A. Sadeghi, and M. Volkamer, "Presenting A Trustable E-Government Voting Management Framework Using TPM. 2010. Online Voting".n.d. https://www.cosic.esat.kuleuven.be/wissec2006/papers/17.pdf on 10th April, 2014.

Yeboah. A, Journal of Information Engineering and Application, Vol. 3, No. 1. 2013.

Ofori-Dwumfuo and E. Paatey. "Electronic Voting in Ghana: Is this Really The Solution To Ghana's Perceived Electoral Challenges After Biometric Registration?", Research Journal of Information Technology 3(2): 91-98, ISSN: 2041-3114. 2011.

The official website of the Independent Electoral Commission of South Africa, http://www.ec.gov.gh/, 2014

Heitmeyer, C.L., Archer, M.M., Leonard, E.I., McLean, J.D., "The Evaluation of Voting Technology". IEEE Transactions on Software Engineering (1), 34, pp. 82–98, 2014.

ES&S Inc, Election Systems & Software: iVotronicTMVoting System, version 9.1.x Election Day Operations Checklist, 2010.

Jones, D.W.,. "Applying formal methods to a certifiably secure software system", Chap 1. Towards the Advances in Information Security. Kluwer Academic, pp. 3–16, 2010

Juels, A., "Proving coercion-resistance of scantegrity II." In: WPES: Proceedings of the 2015 ACM Workshop on Privacy in the Electronic Society, ACM, New York, NY, USA, pp. 61–70.

Küsters, R., Truderung, T., Vogt, A., "Coercion-resistant electronic elections", ICICS, Lecture Notes in Computer Science. Springer, pp. 281–295, 2010

Kemmerer, R.A., "Tools and Techniques for the Design and

Systematic Analysis of Real-Time Systems", 37–50. Kohno, T., Stubblefield, A., Rubin, A.D., Wallach, D.S., "Analysis of an electronic voting system." In: IEEE Symposium on Security and Privacy , p. 27, 2009

Kolano, P., "Of Integrating the Formal Methods into the Development Process. IEEE Software 7 (5). Ph.D. thesis, University of California, Santa Barbara), 2009

Kolano, P.Z., Dang, Z., Kemmerer, R.A., "An Analysis of an electronic voting protocol in the applied Pi-calculus.", in Annals of Software Engineering 7 (1–4), 177–210, 2009

Kremer, S., Ryan, M.D., "The Design and Analysis of imminent Real-Time Systems Using the ASTRAL In": Sagiv, M. (Ed.), of the Programming Languages and Systems— Proceedings of the rare 14th European Symposium on Programming (ESOP'15). Lecture Notes in Computer Science, Springer, Edinburgh, UK, pp.186–200, 2015

Lowry, M., Dvorak, D., "Towards Communicating trust in e-commerce interactions". IEEE Intelligent Systems 13 (5), 45–49, 2008 Ajzen, I. & Fishbein, M. (1972) Attitudes and normative beliefs as factors influencing intentions. Journal of Personality and Social Psychology, 21, 1–9.

Bélanger, F. & Hiller, J. (2005) "In the advent of A framework for collective e-government:" Privacy implications of the Business Process Management Journal, 14, (in press).

Bélanger, F., Hiller & Smith, W. (2012) "Trustworthiness in the electronic commerce industry: the role of privacy, security, and site attributes." Journal, 2–5.

Carter, L. & Bélanger, F. (2013) Mixing the Diffusion of Innovation and Citizen Adoption of E-government Services in The Proceedings of the 1st International E-Services Workshop, 57–63.

Carter, L. & Bélanger, F. (2004) "A Citizen Adoption of E-government Initiatives" in the Proceedings of the 37th Hawaiian International Conference on Systems Sciences, 5–8.

Chadwick, S. (2001) "Analytic Verification of Flight Software." In Management Communication Quarterly, 17, 653–658.

Chau, P. (2013) "in the wake of An empirical assessment of a modified technological acceptance model. Journal of Management Information Systems,13, 185–204.

Cronbach, L. (2007) "The sole

Essentials of Psychology Testing.” Harper & Row, New York, USA.

Doll, W., Hendrickson, A. & Deng, X. (2008) “Using Davis's perceived usefulness and ease-of-use instrument for decision making: a confirmatory and multigroup invariance analysis.” Decision Sciences journal, 29, 839–869.

Martinelli, F., “ The Symbolic inference systems” published in the Proceedings of the 27th International Symposium on Mathematical Foundations of Computer Science, Springer-Verlag, London, UK, pp. 519–531, 2008.ProjectReport,2007

Optimal Key Size of the AVK for Symmetric Key Encryption

Shaligram Prajapat[1] &

Dr. Ramjeevan Singh Thakur[2]

[1] International Institute of Professional Studies, Devi Ahilya University, Indore, India.

[2] Department of Mathematics and Computer Applications, Maulana Azad National Institute of Technology, India

(shaligram.prajapat@gmail.com, ramthakur2000@gmail.com)

Abstract: The security of AVK based cryptosystem can be enhanced merely by exchanging the key using parameters. Today, the major challenge we face in design of AVK model of symmetric key encryption is fixing key length for AVK. On deeper scrutiny, it was revealed that a key of shorter length increases vulnerability of the system. On the other hand, key length beyond optimum length involves unnecessary overheads (suboptimum utilization of bandwidth). Thus, this paper resolves the conundrum of research questions, and answers estimation of optimum key size for AVK model. The paper provides useful insights towards decision making for optimal key length.

Index Terms—AVK (Automatic variable key), Symmetric Key.

I. Introduction

The security of information transmitted over a public network depends upon 3 characteristics namely: (1) Enciphering algorithm (2) Protocol and (3) Key. It is often assumed that a system is secure if we have a strong encryption algorithm or a secure protocol. But, security of the whole cryptosystem may be compromised if the key is mishandled: If somehow opponents get hold on keys, the security of information is compromised. So, a systematic and appropriate use of keys is essential to ensure security of information. In principle, any cryptic key can have following features: Key-Length, Key-Randomness, Key-Lifetime, and Key-Secrecy. Although the dynamism, randomness or variability in key increases security of information transmitted over the communication channel, a number of techniques and methods are still under investigation for its efficient implementation.

The strength of traditional crypto-algorithm is the function of key length which takes longer computation time for large key length. This is an overhead associated with Key Length. Alternative approach for ensuring security is use dynamic keys or fix up the key length and vary it for every new session. But, the major issue would be: if key is kept fixed then what should be its length? Shorter key length may be easy to predict and higher the key would lead to overheads similar to big key size .In the subsequent sections, the paper highlights alternative approaches for the issue. It would also recommend time variant key approaches for better safety. The paper also finds answers for key size of AVK to balance the vulnerabilities and computation time.

II. Related Work

Although numerous literature works are available on key based algorithm for securing and comparing efficiency of information, rare amount of work is available on guidelines for choosing Key-size for AVK based cryptosystem. The documentation [1] of Microsoft insinuates that the chance for success of systematic attacks (where intruder tries each permutation of the key until the desired key to decipher the message is explored) depends on the key size. According to this document, the best alternative for minimizing the success rate of brute force attacks are: (1) Select small key-Lifetime or (2) Select key- size, which is large in length. In the former approach, the smaller key-lifetime minimizes the probability of attacks (even if one of the keys is known). This leads to conclusion of key variability. On the other hand, limitation of key selection in fixed key of longer size is used to minimize the probability of successful attacks by increasing the number of combinations [1].

Almost 13 years back, in 2002, Hellman highlighted the effects of increasing the key-length, for improving security level on the vulnerable network. In his article, the suggestion of suitable key-length for secure information exchange was claimed to be of 90 bits with accordance of trends of increasing key size. As indicated in the Table 1.0. The information exchange with key of larger length is assumed to be more secure due to large computation time requirements. One can straight forward infer that, to secure data we need to increase the key length [2].This is a limitation and impractical aspect from future computing perspective.

According to Moore’s law, the computing power of personal computer doubles approximately in every 18 months. If we check key-length problem in alignment with Moore’s law, the effect of proliferating computing power can be experienced on computation of key size of large lengths. The availability of fast multi-core processors and low cost hardware will equip the cryptanalyst. So, intruders and hackers can rigorously develop new techniques and algorithms to exploit and improve the efficiency of key search to breach system in less time. The estimated time for successful key search attacks must be revised as computing power and resource availability increases.

Table 1. Key Size and its impacts on time and speed

Key Size (in bits)	Significance	Traditional-Time and Present speed
40 bits with 2 ^{40} permutations	1 PC with rate of 1 million keys per second, will take 13 days to try out all possible keys	13 days and in Hours rather than in days
56 bits with 2 ^{56} permutations	DES-with supercomputer of entry rate 92 billion keys per second decrypted the message in 56 hours after trying about 25 percent of the possible keys.	Remined safe for 56 hours and after six months in 22 hrs.An improved version of it was decoded by supercomputer (56-bit DES encrypted message)in about 22 hours,.
64 bits with 2 ^{64} permutations	Better Performance over 56 bit key size	Relatively more time is required with respect to traditional DES,generally provides strong protection against brute force attacks.
128 bits with 2 ^{128} permutations	10 million PC trying 100 billion keys per second will take about 10 ^{13} years to try every possible 128-bit key value.	1000 times longer than the estimated age of the universe (15 billion to 20 billion years).But, symmetric keys that are 128 bits or longer are considered unbreakable by brute force attacks.
192 bits with 2 ^{192} permutations	Excellent performance	Relatively hard and secure
256-bit with 2 ^{256} permutations	Excellent performance	Hard but Highly Secure

In traditional approach, user chooses/creates a key containing a string of characters. The key string may be in the form of alphanumeric, numeric, special symbols etc. Depending upon type of implementation checked by source or destination computer, if the supplied key matches with the one which is associated with the actual user's resource (files, databases, etc), access is granted to all resources belonging to the authorized user.

Primary approach for inexpensive key design is choosing a relatively short string of characters and allowing the user to decide the key, in such a way that the selected key is memorable. However, with such type of keys are easier to guess. Likewise, if user selects a key arduous to remember, it is most probable that he will save the key somewhere (either electronically in hard disk or non-electronically in paper slips).In both cases, the system is equally vulnerable to attack.

An alternate approach to this problem can be implemented by increasing the key length. This would make the system relatively more secure to stand against exhaustive cryptanalyst search. The expected "safe-time" and "breaking-threshold" can prevent(secure) the key from brute force attack. It can be computed by expression (1) and (2). It can also be used to indicate effectiveness of key by its length (used in a given system) [3].

Safe Time: It is the maximum time required to guess a key (in a brute force attack) and is computed using the formula (1):

Safe time =0.5*Total number of possible keys*Time to enter one key (1)

Breaking Threshold: It is the optimum time taken to find the right key to breach the system. For a given key, selected from characters set domain of size N, breaking threshold for a given key is given by:

Breaking threshold = 1/2*N^x*L/(R) (2)

Where: X = length of key (in number of characters), R = Data entry and transmission rate: R characters per minute. N = Size of character set domain i.e. number of letters, numbers, and special symbols (from which the key is selected). Thus, Number of characters involved for entry and replying in a login attempt is N characters.

III. Optimum Key for AVK Model

The AVK model of variability of key for symmetric key has already been proposed in the literature for improved security aspects. In AVK approach, key is varied in respect of time over sessions. Initially key is generated by variable information (exchanged in prior session).The superiority of AVK over a fixed key or key with variable length has been proved in literatures [4, 5, 6, and 7]. Fibonacci–Q matrix, Sparse matrix based approaches are recent approaches that can be used to generate variable keys. Dynamic keys can be generated among a number of users. [8]. A number of techniques to generate time variant key are proposed. In reference [7], we studied a comparative study performed to find out the best techniques among different key generation techniques.

IV. Experimental Setup & Results

A Python 2.7.8 script using pandas library and matplotlib library were used to plot the result under various parameter of x and y. The Pandas library was used to compute the data using the formula (2).The calculated value of time to guess (in Hrs) was stored with corresponding key size in a data frame. The plot ( ) function was used to demonstrate the result.

The effect of increasing the key size was investigated for finding out what would be the optimum key length. It is assumed that data entry rate for entering key is 120 characters per minutes, the character set size is 20 characters (Frequent characters are not considered in character set) login length is 15 characters. Using the Anderson’s formula,

N^x≥(4.32*104 *T* M)/(L* p0) (3)

In (3) it is also assumed that the probability for a correct guess is p and the time period in months for systematic attack has been made 24 X 7 in M months, and the lower bound probability is p0 .This expression can be used to decide length of the key (x). Such that it reduces the possibility middle attack as compared to p0.

The Effect of increasing the key length is shown in Table 2. The comprehensive security was studied for various character lengths. The plotted graph demonstrates that the optimum key length lies in between 4 to 8.

Table 2 Effect of increasing key length

Size of key (in characters)	**Time taken to successful guess (In Hrs.)**
20	7.610350076
26	36.73370624
52	2350.957199
62	6754.213706
72	16566.07562

For plotting the graphs for estimation we have considered the system with: Key length = 6, Login length = 6, Typing speed: 120 characters per minutes, x = Length of character set and Time taken to guess (in Hrs) is y then

$$y = \frac{0.5 * x^6 * 15}{120 * 60} \qquad (2)$$

The plot is shown in Fig. 1.

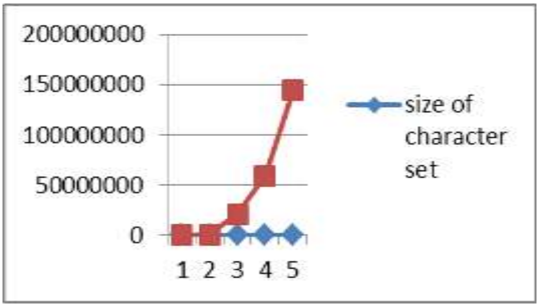


Figure 1. Time to Guess Key vs Length of Character Set

$$y = \frac{0.5 * 72^{x} * 15}{120 * 60} \qquad (4)$$

Assuming length of character set= 72 (including alphanumeric key elements) and login length =15 and typing speed: 120 characters per minutes. The plot of "key length" with respect to "time taken to guess (in Hrs)" is presented in Fig. 2.

Table 2. Key Length V/S Time

Key Length	Time to guess (in Hrs)
1	0.020833333
2	0.416666667
3	8.333333333
4	166.6666667
5	3333.333333
6	66666.66667
7	1333333.333
8	26666666.67
9	533333333.3
10	10666666667
11	2.13333E+11
12	4.26667E+12
13	8.53333E+13
14	1.70667E+15
16	6.82667E+17
20	1.09227E+23
25	3.49525E+29
50	1.17281E+62
100	1.3205E+127=

The graph indicating key length and time to guess the key seems linear, assuming length of key i.e. number of characters on X axis and time to guess all permutation (in hours) is as shown below in (Fig.2).

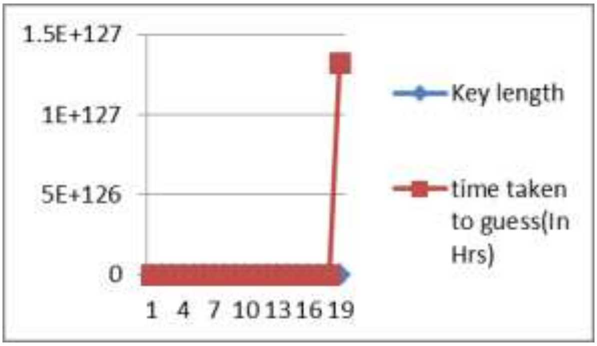


Figure 2. Time to Guess Key vs Length of Character Set

The plot of Fig.2 displays homogeneous behavior over all key size. So, a deeper introspection look is required to observe its effect over time.

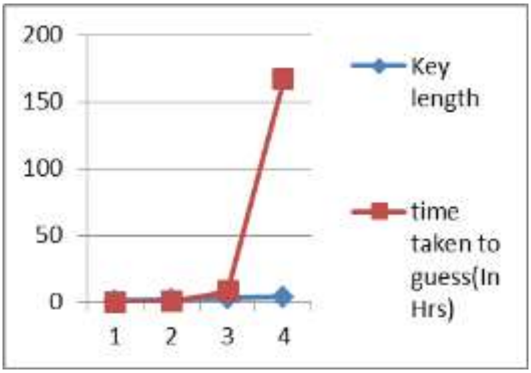


Figure 3. Time to Guess Key vs Length of Character Set

Now to explore the effect of time (to guess a key) for Key length varied from 4 to 8 characters , as per the plot in Fig. 3.

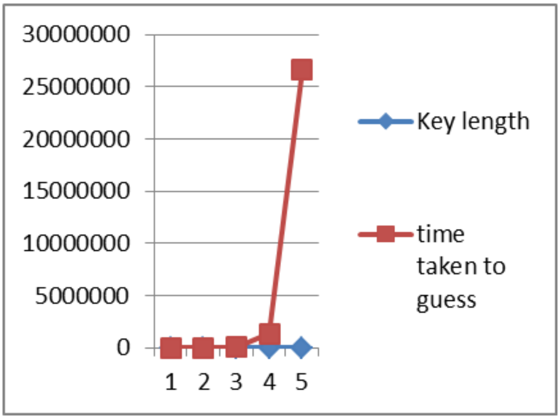


Figure 4. Time to Guess Key vs Length of Character Set
Effect of time to guess a key for Key length varied from 6 to 10 characters.

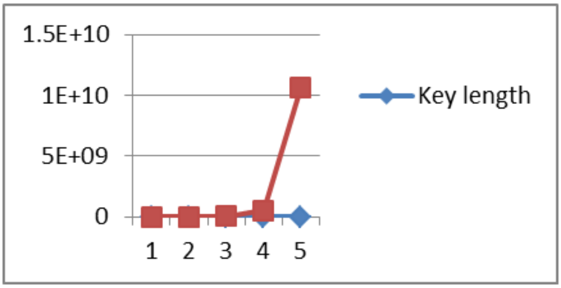


Figure 5. Time to Guess Key vs Length of Character Set
Effect of time to guess a key for Key length varied from 8 to 14 characters.

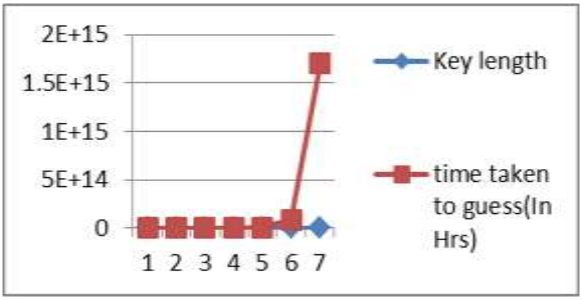


Figure 6. Time to Guess Key vs Length of Character Set
Effect of time to guess a key for Key length varied from 12 to 16 characters.

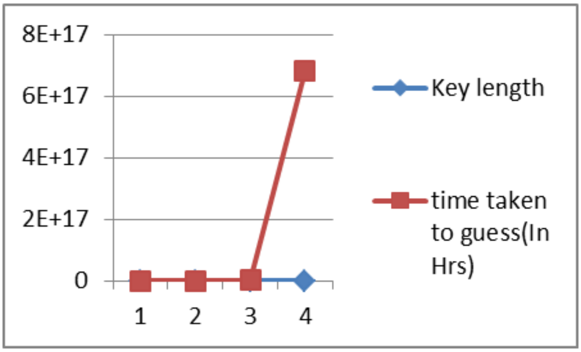


Figure 7. Time to Guess Key vs Length of Character Set

Effect of time to guess a key for Key length varied from 16 to 25 characters.

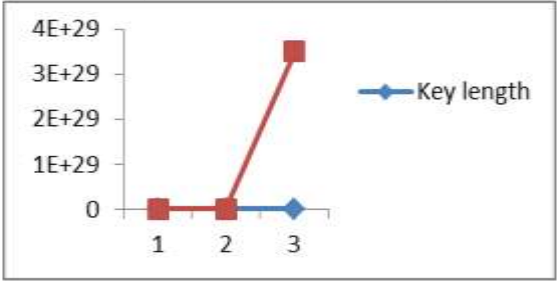


Figure 8. Time to guess key vs length of Character set

For implementation of Automatic Variable key with fixed length but dynamic in nature from session to session should be at least 5 or 6 characters for sufficient resistance against vulnerability.

V. Conclusions

In traditional cryptosystem the practical difficulty of increasing the key length to reduce the probability of damage from side channel attacks, makes it arduous and inconvenient to remember long string. User has to record the key either in the system file or on piece of paper, both are undesirable. Relatively shorter keys would be easy to remember and convenient in handling and changing from session to session from the perspective of AVK based cryptosystems. The work described in this paper point outs that optimal key length for fixing up with automatic variable key length of 5 or 6 characters (key length to prevent from system attack) is sufficient for a session. Once a key of this key length is initiated then it can be changed from session to session, so that the security of the system is not compromised. It is also worth mentioning that key generally does not fail because of brute force attack but, fails as a result of mishandling of the key.

VI. Acknowledgement

This work is supported by research

project under Fast Track Scheme for Young Scientist from DST, New Delhi, India. Scheme 2011-12, No. SR/FTP/ETA-121/ 2011 (SERB), dated 18/12/2012References.

References

BlueKrypt, Cryptographic Key length Recommendation http://www.keylength.com/en/4

Martin E. Hellman, "An overview of public key cryptography", IEEE Communication Magazine- May 202

Lane, "Security of Computer based Information System", Macmillan series,1990.

R.Goswami, S. Chakraborty, A. Bhunia,C. T. Bhunia ,'Generation of Automatic Variable Key under Various Approaches in Cryptography System", Journal of The Institution of Engineers (India): December 2013, Volume 94, Issue 4, pp 215-220

Galen E. Pickard, Roger I. Khazan, Benjamin W. Fuller, Joseph A. Cooley, "DSKE: Dynamic Set Key Encryption", MIT Lincoln Laboratory.

Shaligram Prajapat, Sachin Saxena, Amber Jain," Implementation of Information Security with Fibonacci Q-Matrix", in the International Conference on Intelligent Computing and Information System (ICICS-2012).

Shaligram Prajapat, Amber jain, R.S.Thakur, "A Novel Approach For Information Security with Automatic Variable Key Using Fibonacci Q-Matrix", International Journal of Computer & Communication Technology (IJCCT) ISSN (ONLINE): 2231 - 0371 ISSN (PRINT): 0975 – 7449 Vol-3, Iss-3, 2012, p.p. No. 54-57.

Shaligram Prajapat, Dr. R.S. Thakur et al., "Sparse approach for realizing AVK for Symmetric Key Encryption", International Research Conference on Engineering, Science and Management 2014 (IRCESM 2014) Dubai, UAE –ISBN 978-93-83303-51-9

Shaligram Prajapat, Dr. R.S. Thakur, "Time variant approach towards symmetric key", SAI-Conference London, cosponsored by IEEE and Springer , October-2013.

Author's profiles:

Shaligram Prajapat, He is pursuing Ph.D. under supervision of Dr.

Ramjeevan Singh Thakur, from Department of Mathematics and Computer Applications, Maulana Azad National Institute of Technology.He has received B.Sc.(Electronics), M.Sc.(Com.Sc.), M.Tech(Com.Sc.), M.Phil.(Comp.Sc.) from Devi Ahilya University,Indore.He is International Institute of Professional Studies (IIPS) ,Devi Ahilya University Indore, as Reader for MCA and M.Tech Courses since 2007.With over 15 years of teaching experience, He has reviewed five international books of Pearson education, 3 papers in international journals including Springer and Atlantis press. He has also presented paper in international and national conferences. He is member of various professional bodies like IEEE, ISTE, ACM, CSI, CSTA, IAENG, IEEE(Computer Society).

Dr. Ramjeevan Singh Thakur is eminent researcher and Associate Professor in the Department of Computer Applications at Maulana Azad National Institute of Technology, Bhopal, India. He is a Teacher, Researcher and Consultant in the field of Computer Science and Information Technology. He earned his Master Degree from Samrat Ashok Technology Institute, Vidisha (M.P.) in 1999. And Ph.D. Degree (Computer Science) From Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.) in 2008. He has published more than 75 Research Paper in National, International, Journals and Conferences. He has visited several Universities in USA, Hong Kong, Iran, Thiland, Malaysia, and Singapore. His areas of interest include Data Mining, Data Warehousing, Web Mining, Text Mining, and Natural Language Processing. He has also received DST Young Scientist Award-2011 in Engineering under Fast Track Scheme, Department of Science & Technology, New Delhi, India. His area of interest are Data Mining, NLP Bioinformatics, Soft Computing