Emergent Future Situation Awareness: A Temporal Probabilistic Reasoning In The Absence of Domain Experts

Isaac Osunmakinde, *SMIEEE* and Antoine Bagula
Department of Computer Science, Faculty of Sciences, University of Cape Town,
18 University Avenue, Rhodes Gift, 7707 Rondebosch, Cape Town/South Africa
{segun, bagula}@cs.uct.ac.za

Abstract. Dynamic Bayesian networks (DBNs) are temporal probabilistic models for reasoning over time which are rapidly gaining popularity in modern Artificial Intelligence (AI) for planning. A number of Hidden Markov Model (HMM) representations of dynamic Bayesian networks with different characteristics have been developed. However, the varieties of DBNs have obviously opened up challenging problems of how to choose the most suitable model for specific real life applications especially by non-expert practitioners. Problem of convergence over wider time steps is also challenging. Finding solutions to these challenges is difficult. In this paper, we propose a new probabilistic modeling called Emergent Future Situation Awareness (EFSA) which predicts trends over future time steps to mitigate the worries of choosing a DBN model type and avoid convergence problems when predicting over wider time steps. Its prediction strategy is based on the automatic emergence of temporal models over two dimensional (2D) time steps from historical Multivariate Time Series (MTS). Using real life publicly available MTS data on a number of comparative evaluations, our experimental results show that EFSA outperforms popular HMM and logistic regression models. This excellent performance suggests its wider application in research and industries.

Keywords: Dynamic Bayesian Networks, Situation Awareness, Prediction, Multivariate Time Series.

1 Introduction

Industrial practitioners and researchers observe multivariate time series (MTS) from their daily business activities or dynamical systems (e.g. medical systems, retail, sensor networks, etc). Complex hidden relationships (or patterns) are often embedded among the variables that describe such activities within and across the time steps. Some classical methods such as neural networks and statistical logistic regression models have been applied to predict such hidden patterns but they fall short of proving their prediction capabilities [1]. Using more sophisticated approaches, these hidden patterns can be revealed from the historical MTS to predict risks, or guide actions to be taken at particular future times. Any retail business may for instance intend to know which products require declaration of discounts among selected outlets in specific months for next year. This is an example of prediction over time within a multitude of complex situations and dynamic Bayesian networks are well suited for reasoning over time in complex environments [2] [3].

The Hidden Markov Model (HMM) is a common and most simple form of DBNs which has gained its wide applicability in speech recognition [3] [4]. Recently, researchers have developed many HMM representations of dynamic Bayesian networks with different characteristics. Murphy [2] proposed some variants of HMM as explicit representations of DBN such as: hierarchical HMM, coupled HMM and factorial HMM. Shenoy [5] presented another DBN model for Brain-Computer Interfaces. As experts, they explicitly modeled the hidden network structure and dependencies between different brain states.

However, the varieties of DBNs have obviously opened up challenging problems of how to choose the most suitable model for various real life applications. Some prediction models also suffer from convergence or exponential problems over wider time steps [3]. That is, the prediction steps get stuck towards zero or tend towards infinity. To complicate the situation further, the challenges have made technologies such as the DBNs too complex for non-experts including practitioners [6] (seasoned software programmers, managers, etc).

In an attempt to address some of these challenges, Deviren [4] presented Structural learning of DBN and also applied it to speech recognition. Their DBN was learnt under a number of assumptions from experts. For instance, they observed stationary assumption which made their DBN leads to a repeated network structure at each time step. In reality, situations in some time steps may change. This is evident that most of these existing DBNs approximate their models. That is, they do not truly (or completely) emerge the network structures and probability distributions but the basis of DBN requires modeling both [2] [3]. Murphy [2] confirms that the HMM, which is the basis of most of these representations of the existing DBNs, are limited in their expressive power. Finding solutions to these challenges is difficult.

In this paper, we propose a new probabilistic modeling called Emergent future Situation Awareness (EFSA) technology which predicts trends over finite future time steps. The EFSA eliminates the worries of choosing a good DBN model and avoids convergence problems. Its automatic and complete emergence of temporal models (network structure and probability distributions) over time from historical MTS is the strategy of the prediction. Our major contributions are as follows:

- The derivation of a temporal probabilistic theory for the new EFSA technology which predicts trends over future time steps in the absence of domain experts.
- The development of the EFSA algorithm which facilitates the mitigation of the worries of practitioners and researchers for choosing a DBN model from the multitude of varieties for specific applications.
- Using a 2D strategy to avoid convergence problems when predicting longer time steps, our EFSA model supports wider applicability for all users: experts and non experts.

The rest of the paper is organized as follows. In section 2, we present the theoretical backgrounds of the dynamic Bayesian networks and our previous ESA (emergent situation awareness) technology. The details of the proposed EFSA technology are presented in section 3. In section 4, we evaluate the performance of the EFSA's consistency and accuracy, and benchmark it with the HMM and the logistic regression models. This paper concludes in section 5.

2 Theoretical Background

2.1 Dynamic Bayesian Networks (DBNs)

Dynamic Bayesian networks are temporal probabilistic models which are often referred to as an extension of the Bayesian network (BN) models in artificial intelligence [2] [3]. A Bayesian belief network is formally defined as a directed acyclic graph (DAG) represented as G = {X(G), A(G)}, where X(G) = $\{X_1,\ldots,X_n\}$, vertices (variables) of the graph G and $A(G) \subseteq X(G) \times X(G)$, set of arcs of G. The network requires discrete random values such that if there exists random variables X_1, . . ., X_n with each having a set of some values x_1, . . ., x_n then, their joint probability density distribution is defined in equation 1;

$$pr(X_1,\ldots,X_n) = \prod_{i=0}^{n} pr(X_i \mid \pi(X_i)) \qquad (1)$$

where $\pi(X_i)$ represents a set of probabilistic parent(s) of child X_i [3]. A parent variable otherwise referred to as *cause* has a dependency with a child variable known as *effect*. Every variable *X* with a combination of parent(s) values on the graph G captures probabilistic knowledge (distribution) as a conditional probability table (CPT). A variable without a parent encodes a marginal probability.

However, the inability of the BNs to capture time as temporal dependencies facilitated the developments of various ways of modelling the dynamic Bayesian networks presented at the introduction. The variables and the CPTs of the BNs are similar to the states and the probabilities used in the temporal dependencies of the DBNs. According to [3], a DBN is suitable for modelling environment that emerges (changes) over time and has the capability to predict future behaviour of the environment. Any DBN observes the first-order of Markov model which states that, future event V_{t+1} is independent of the past given the present V_t [3]. Since DBN handles complex situations of multiple dependent events of Markov model over time, researchers [2] [3] present the following three parameters required to construct a DBN model: prior matrix, *Pr* (V_0); transition matrix, *Pr* $(V_t \mid V_{t-1})$; and sensor matrix, *Pr* $(E_t \mid V_t)$. The prior matrix defines the initial probability distribution of states V_0 at the start of emergence of DBNs. The transition matrix describes time dependencies for the transitions of DBN states V_t. Also, the sensor matrix captures the probabilistic distributions from the relationships of observation variables E_t at any time step.

In conjunction with the DBN matrices, equation 3 shows the combined joint probability distribution for any temporal model up to a finite time *t*.

$$\Pr(V_0, V_1,\ldots,V_t, E_1,\ldots,E_t) = \Pr(V_0) \prod_{i=1}^{t} \Pr(V_i \mid V_{i-1}) \Pr(E_i \mid V_i) \qquad (2)$$

The emergence of our DBN technology is based on the theoretical principles underpinning situation awareness [9] in order to make anticipatory planning.

2.3 The ESA Technology

The ESA [13] is an innovative technology, which completely emerges temporal models and reveals the hidden behavior of what is currently happening over time in any domain of interest. Formally, let $\{V^t, E^t\}$ represent the set of state and observed DBN variables in ESA at time *t.* The DBNs are emerged over all the non-negative current time steps $t \in T$, such that $T = \{t_1, t_2 \ldots t_n\}$ and the interlinked probabilistic relationships at each time step *t* is represented in equation 3. Equation 3 represents the interconnections of changing networks (or frames) and probability distributions over the time steps.

$$pr(V_1^{t_1}, V_2^{t_1} \ldots, V_m^{t_1}) \overset{\Delta}{\equiv} pr(V_1^{t_2}, V_2^{t_2} \ldots, V_m^{t_2}) \overset{\Delta}{\equiv} pr(V_1^{t_n}, V_2^{t_n} \ldots, V_m^{t_n}) \qquad (3)$$

where $\overset{\Delta}{\equiv}$ implies equivalence is *not* true generally. The attractive performance of the ESA encourages its successful applications in many areas, most notably in project management [11]. On the other hand, the ESA falls short of predicting into the future. We therefore conjecture that a variant of the ESA called the EFSA is required to efficiently predict into the future based on the historical time steps.

3 The Proposed EFSA Technology

3.1 Theoretical Derivations of the EFSA

Researchers [3] assert that prediction too far (wider time lag) into the future converges to a fixed point (i.e. remains constant for all time). In order to minimize the convergence problem, the EFSA predicts future trends using the strategy of a two dimensional (2D) time steps. The first dimensional space of time steps $\{t_1, t_2 \ldots t_n\}$ monitors the behavioural current patterns as used in the ESA. The second dimensional time steps $\{T_1, \ldots, T_m\}$ observes the historical patterns for each of the time steps $\{t_1, t_2 \ldots t_n\}$. This is an extension of any period *T* in the ESA. Therefore in practice, at the end of the tn of Tm, the EFSA updates further future trends to ensure accuracy. Let the DBN variables V^t span the space of 2D time steps represented in the system of equations 4.

$$T_1 = \{ V_i^{t_1}, V_j^{t_2} \ldots, V_\alpha^{t_n} \};\ T_2 = \{ V_i^{t_1}, V_j^{t_2} \ldots, V_\alpha^{t_n} \};\ \ldots;\ T_m = \{ V_i^{t_1}, V_j^{t_2} \ldots, V_\alpha^{t_n} \}$$

for each $i, j, \alpha = 1,2,\ldots\ell$ (4)

ℓ is the length of the DBN variables and *m* is the length of the history. All the changing parameterizations (the DAGs and the probability distributions) of the DBN in the EFSA are now carried out across the historical time steps $T_1 \ldots T_m$. That is, the emergence (or learning of the temporal models) takes place across the links:

$$\{ V_i^{t_1}, V_i^{t_1} \ldots, V_i^{t_1} \}, \ldots, \{ V_\alpha^{t_n}, V_\alpha^{t_n} \ldots, V_\alpha^{t_n} \}.$$

Once the temporal probabilistic model emerges, prediction with reasoning now acts on the model. From Markovian principle [3] which states that next states of a system depend on the finite history of the previous states, we can now have multiple *n* predictions from the space of 2D time steps in equation 4 into the future as follows:

$$T_{m+\lambda} = \{ V_i^{t_{1+\lambda}}, V_j^{t_{2+\lambda}} \ldots, V_\alpha^{t_{n+\lambda}} \},$$

$$\Rightarrow \Pr(V_i^{t_{1+\lambda}}) = \Pr(V_i^{t_{1+\lambda}} \mid E_i^{1:t_1}); \Pr(V_j^{t_{2+\lambda}}) = \Pr(V_j^{t_{2+\lambda}} \mid E_j^{1:t_2}); \ldots;$$

$$\Pr(V_\alpha^{t_{n+\lambda}}) = \Pr(V_\alpha^{t_{n+\lambda}} \mid E_\alpha^{1:t_n}), \quad \text{for some } \lambda > 0 \tag{5}$$

In equation 5, E_i, E_j,…E_α are the set of evidences or observations of V_i, V_j,…V_α respectively made so far within the space of time steps. Equation 5 is therefore the set of predictions that can be computed by the Bayesian inference algorithms such as Variable elimination, etc [3]. The variable elimination implemented in [10] was integrated as the inference engine in the EFSA due to its efficiency. Therefore, the EFSA performs a multiple future predictions from the space of time.

3.2 The Description of the EFSA

The emergence of DBN or temporal probabilistic model of the EFSA is often a task of BN learning algorithms provided it can learn across the time steps [2]. Therefore, the outlines of learning DBNs automatically from EFSA algorithm are now refined from [13] as follows:

INPUT (D_s : Multivariate Time Series - MTS)

---

1. While D_s = MTS,
   - [i] Set t, the frame count, to 1.
   - [ii] Set T, the historical time step, to 1, 2, …, m
   - [iii] Let $d_t \in D_s$, $\forall$ $t = 1, 2, \ldots, n$.
   - [iv] For each $t <= n$,
   - [v] For each T <= m,
     - Select frame d_t into { d_t }
   - [vi] Increment T by 1.
     - Invoke Learning_Algorithms ({d_t}).
     - Store the emerged frame in *n by m* matrix *B*.
   - [vii] Increment t by 1.
2. Return the DBN in B
3. Predict the next *n* time steps using inference engine, then exit.

Figure 1: Emergent Future Situation Awareness Algorithm

In Figure 1, all parameters retain their usual meanings and d_t is a frame dataset at time *t*. It is selected into set {d_t} over *T* for learning frames of the DBN. Any Bayesian learning algorithms such as [7] [8] can be used as a subroutine. The algorithms carry out the intra-slice and inter-slice learning over time. Each variable in step *T* must have parents in step *T-1*. We integrated the genetic algorithm [8] to learn the DBNs due to its efficiency.

4. Performance Evaluation

One of the objectives of our proposed EFSA technology is to bring theory to practice (implementation) with an emphasis on applications and practical work. The HMM and logistic regression models have been used in our experiments as a baseline of comparison with our EFSA model. The logistic regression model is a function of dependent variable over the independent variables [1].

4.1 Experiment 1: Comparing EFSA consistency with other popular models

Our intention here was to determine whether the EFSA can predict multiple n-time steps consistently. As a proof of concept, we carried out the evaluations of the three models on three MTS datasets - DIABETES and SENSOR datasets from UCI repository [12], and a real life RAINFALL dataset obtained from a Southern African country (Botswana). The treatment records from the behavior of a diabetes patient were captured electronically as MTS. It contains several treatment measurement codes such as 33 (regular insulin dose), 48 (unspecified blood glucose), etc. It is expected here to predict treatment measurements required in future for the patient based on the historical behavioral patterns. For instance, equation (6) below is a situation which predicts how much of the minimum (about 7 units) measurement of regular insulin dose will be required for the next 12 months in the year 1991.

$$Pr(Measurement^{t+\lambda} <= 7 \text{ units} \mid Code^{t} = 33) \tag{6}$$

for all $t \subset T$, where in diabetes MTS, $t = \{Jan\ldots Dec\}$ and $T = \{1988, 1989, 1990\}$. A common empirical technique to evaluate the performance of Bayesian network technologies is to use a basic cross validation [3]. We adopted the cross validation approach by setting 1991 time step as actual test data and learnt (or trained) the DBN model across 1988 to 1990 time steps. The EFSA reasons with the temporal model and acts by predicting over time as described in equation (6). This experiment was repeated using basic HMM constructed dynamically on the fly using GeNle Bayesian software [13]. Similarly, equation (6) was also repeated using the logistic regression model implemented in R statistical software [14]. The actual and the predicted results were recorded in each experiment and are shown in Table 1.

The sensor dataset captures the traffic of people flowing in and out of main door of a CalIt2 building at UCI. Our objective here is to be able to predict counts of people for every half an hour over future weeks based on the historical behavioural patterns of traffic. For instance in equation (7), we want to predict the possibility of counting average number (between 8 and 17) of people that flows out of the building for the next 5 weeks.

$$Pr(Count^{t+\lambda} = \text{'}8<=17\text{' people} \mid Flow^{t} = \text{'outflow'}) \tag{7}$$

for all $t \subset T$, where in Sensor MTS, $t = \{Week\text{-}1\ldots Week\text{-}5\}$ and $T = \{July, Aug, Sept\}$. We also adopted the cross validation approach by setting October time step as actual test data and learnt (or trained) the DBN model across 15 weeks of July to Sept time steps. The EFSA reasons with the temporal model and acts by predicting over time as described in equation 7. This experiment was also repeated using other models. The actual and the predicted results were recorded in each experiment and are shown in Table 1.

Table 1: Performance Comparison of Future Predictions on Three Situations Among EFSA, HMM and Logistic Regression Model

Data Sets	Time Steps	Actual (%)	EFSA (%)	HMM (%)	Logistic(%) Regression
Diabetes	Jan	40.15	73.26	70.43	87.30
	Feb	50.79	58.7	72.45	87.31
	Mar	74.64	90.99	79.45	87.41
	Apr	65.42	80.44	89.69	87.42
	May	72.51	69.08	89.07	87.46
	Jun	60.14	59.69	90.16	87.48
	July	69.43	76.54	93.28	87.51
	Aug	61.09	75.69	95.31	87.53
	Sept	55.17	85.48	95.36	87.56
Sensor	Wk-1	19.55	12.46	24.13	25.29
	Wk-2	20.88	13.39	23.07	26.18
	Wk-3	24.44	18.69	33.17	37.07
	Wk-4	27.11	18.81	33.41	37.95
	Wk-5	3.11	2.76	21.01	38.01
Rainfall	Jan	70.22	53.57	58.24	45.88
	Feb	72.75	56.67	51.12	45.62
	Mar	72.39	55.48	44.09	45.33
	Apr	61.95	55.29	41.20	45.04
	May	62.34	56.67	40.02	44.72
	Jun	78.21	62.67	39.59	44.38
	Jul	85.73	59.81	38.90	44.03
	Aug	78.85	58.38	38.89	43.65
	Sept	83.45	58.00	36.87	43.25
	Oct	80.30	60.05	25.65	42.82
	Nov	89.21	62.31	25.14	42.38
	Dec	91.26	65.05	24.69	41.90

The real life rainfall dataset was obtained from a Southern African country (Botswana) to access onsets of rainfall for Farmers to understand their varying planting dates. Our objective here is to be able to predict normal onset at any station over future months in every coming year. For instance, equation 8 predicts the normal onset of rainfall over future months for a given station number 2. This may include more complex conditions, like considering how sea anomalies affect the onset and wind, as shown in equation 9 which other methods such as regression model struggle to handle [2]. For the purpose of comparison, we keep it simpler as equation 8.

$$Pr(Onset^{t+\lambda} = \text{'normal'} \mid Station^{t} = 2) \quad (8)$$

$$Pr(Onset^{t+\lambda} = \text{'normal'} \mid Station^{t} = 2, Sea\_Anom > 0.5, wind < 7.7units) \quad (9)$$

for all t ⊂ T, where in Rainfall MTS, *t = {Jan…Dec}* and *T = {1971,…2000}.* We also adopted the cross validation approach by setting year 2001 time step as actual test data and learnt (or trained) the DBN model across 1971 to 2000 time steps. The EFSA also reasons with the temporal model and acts by predicting over time as described in equation 8. This experiment was also repeated using HMM and logistic regression models. The actual and the predicted results were recorded in each experiment and are shown in Table 1.

The Table reveals the consistencies or how each model captures the direction of predictive patterns. That is, increase or decrease in predictions from one time step to the next when compared with the actual results. For instance, one can see sensor results in Table 1 as the EFSA prediction increases from 12.46 in week-1 to 13.39 in week-2 and this corresponds to a rise in the actual results. In view of this, one can see generally in Table 1 that the EFSA has the best prediction directions (consistency) of 50%, 100% and 70% for the Diabetes, Sensor and the Rainfall datasets respectively. This results from the fact that the EFSA truly evolves its network and probability distribution with the aid of its 2D strategy of the predictions.

4.2 Experiment 2: Comparing EFSA Accuracy with Other Popular Models

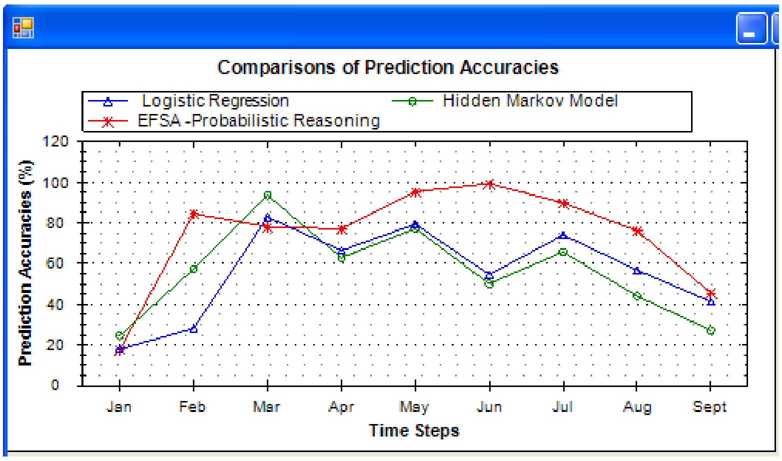

Figure 2: Temporal probabilistic reasoning of the EFSA improves prediction accuracies on Diabetes better than the HMM and the logistic regression model.

We conducted another set of experiments to measure how accurate are the predictions of the models compared to the actual results. The accuracy is simply computed as the difference between a 100% and the percentage error deviation, where the error is the absolute difference between actual value from the test data and the predicted value from the models, divided by the actual value [3]. The prediction accuracies on all datasets are computed from Table 1 and for instance the results of diabetes are specifically recorded in Table 2 accordingly. Figure 2 therefore compares the performance accuracies of the EFSA with other models on Diabetes results in Table

2. The objective here is to improve the prediction accuracy. Observe the predictions of the regression model on diabetes and rainfall in Table 1 as it tends towards a convergence problem.

Over the time steps, the average accuracies of the EFSA, HMM and the regression models on sensor predictions are 72.49%, 62.38% and 50.74% respectively. Also, the average accuracies of the three techniques on rainfall predictions are 76.76%, 51.79% and 57.87% respectively. This shows that the overall accuracy of the EFSA as improved than others is *74.3%*. Thus, the EFSA is more consistent and performs better with future predictions within the multivariate time series.

Table 2: Evaluation of Accuracy for Future Predictions on Diabetes Situations Among EFSA, HMM and Logistic Regression Models

Dataset	**Time Steps**	**EFSA (%)**	**HMM (%)**	**Logistic(%) Regression**
Diabetes	Jan	17.53	24.58	17.90
	Feb	84.43	57.35	28.11
	Mar	78.09	93.56	82.89
	Apr	77.04	62.90	66.37
	May	95.27	77.16	79.38
	Jun	99.25	50.08	54.54
	July	89.76	65.65	73.96
	Aug	76.10	43.98	56.72
	Sept	45.45	27.15	41.29
Average Accuracy		**73.65**	**55.82**	**55.68**

5 Concluding Remarks and Future Work

In this paper, we developed and presented the EFSA technology as a new temporal probabilistic reasoning for consistent multiple predictions into the future in the absence of domain experts. This shows that non experts now have fewer worries in choosing from the multitude of DBN types for real life applications.

This study shows that the EFSA can potentially become a powerful temporal probabilistic model used by both experts and non-experts to predict future trends in anticipatory planning. This technology simply emerges from environments and predicts in any domain of interest. The improved overall 74.3% accuracy of the EFSA over the 56.66% of HMM and 54.76% of logistic regression model when evaluated on the domains of the three datasets guarantees the reliability of the EFSA in many diverse areas. The relative efficiency of the EFSA suggests its wide application to make DBNs much simpler for use by researchers and in industries. We are currently developing an economic scalable model for handling massive MTS for the EFSA.

Acknowledgements. Our appreciation goes to Complex Adaptive Systems (Pty.) Ltd. and the Science Faculty, University of Cape Town for their financial supports.

References

1. Wong, M.L., Leung, K.S.: An efficient data mining method for learning Bayesian networks using an evolutionary algorithm-based hybrid approach. IEEE Transactions on Evolutionary Computation. Vol. 8, Pg 378-404 (2004)
2. Murphy, K.: Dynamic Bayesian networks representation, inference and learning, PhD thesis, UC Berkeley, Computer Science Division (2002)
3. Russell, S., Norvig, P.: *Artificial Intelligence* (A Modern Approach, 2nd Edition, Prentice Hall Series Inc. New Jersey 07458 (2003)
4. Deviren, M., Daoudi, K.: Structural Learning of Dynamic Bayesian Networks in Speech Recognition, Proceedings of Eurospeech, Aalborg, Denmark (2001)
5. Shenoy, P., Rao, R.P.N.: Dynamic Bayesian Networks for Brain-Computer Interfaces, Advances in NIPS, MIT Press, vol. 17 (2005)
6. Silva, E, Plazaola, L., Ekstedt, M.: Strategic Business and IT Alignment: A Prioritized Theory Diagram, *In Proceedings of PICMET*, Turkey (2006)
7. Larranaga, P., Kuijpers, C., Murga, R., Yurramendi, Y.: Learning Bayesian Network Structures by Searching for the Best Ordering with Genetic Algorithms, *IEEE Transactions on Systems, Man, and Cybernetics*, 487-493 (1996)
8. Osunmakinde, I.O., Potgieter, A.: Emergence of Optimal Bayesian Networks from Datasets without Backtracking using an Evolutionary Algorithm. Proceedings of the Third IASTED International Conference on Computational Intelligence, Banff, Alberta, Canada, ACTA Press, ISBN: 978-0-88986-672-0, pp. 46 - 51, (2007)
9. Endsley, M.R.: Theoretical underpinnings of situation awareness: a critical review, Situation Awareness Analysis and measurement, Mahwah, NJ: Lawrence Erlbaum Associates, pp. 3- 32 (2000)
10. Cozman, F.: JavaBayes, Bayesian Networks in Java, University of Sao Paulo (2001), [Online], Available: http://www.cs.cmu.edu/~javabayes/Home/.
11. Balikuddembe, J.K., Osunmakinde, I.O., Potgieter, A.E.: Software Project Profitability Analysis Using Temporal Probabilistic Reasoning, IEEE CS proceedings of the International Conference on Advanced Software Engineering & Its Applications (ASEA), China, accepted for publications (2008)
12. Newman, D., Hettich, S., Blake, C., Merz, C.: UCI Repository of Machine Learning Databases (University of California, Department of Information and Computer Science, Irvine, CA, (1998) http://www.ics.uci.edu/$\sim$mlearn/MLRepository.html
13. GeNle 2.0, Decision Systems Laboratory, University of Pittsburgh (2006) URL = http://genie.sis.pitt.edu
14. R Development Core Team: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0, (2008). URL = http://www.R-project.org.