



An Improved Hybrid Algorithm for Optimizing the Parameters of Hidden Markov Models

**Abukari Abdul Aziz Danaa^{1*}, Mohammed Ibrahim Daabo²
and Alhassan Abdul-Barik³**

¹Department of Computer Science, Tamale Technical University, Tamale, Ghana.

²Department of Computer Science, C. K. Tedam University of Technology and Applied Science, Navrongo, Ghana.

³Department of Cyber Security and Computer Engineering Technology, C. K. Tedam University of Technology and Applied Science, Navrongo, Ghana.

Authors' contributions

This work was carried out in collaboration among all authors. Author AAD did the preliminary analysis and simulations and wrote the first draft of the manuscript with supervision from Authors MID and AAB. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/AJRCOS/2021/v10i130235

Editor(s):

(1) Dr. G. Sudheer, G. V. P College of Engineering for Women, India.

Reviewers:

(1) Adnan Gutub, Umm Al-Qura University, Saudi Arabia.

(2) Balaji V, Vellore Institute of Technology, India.

Complete Peer review History: <http://www.sdiarticle4.com/review-history/69810>

Original Research Article

Received 10 April 2021

Accepted 19 June 2021

Published 25 June 2021

ABSTRACT

Hidden Markov Models (HMMs) have become increasingly popular in the last several years due to the fact that, the models are very rich in mathematical structure and hence can form the theoretical basis for use in a wide range of applications. Various algorithms have been proposed in literature for optimizing the parameters of these models to make them applicable in real-life. However, the performance of these algorithms has remained computationally challenging largely due to slow/premature convergence and their sensitivity to preliminary estimates. In this paper, a hybrid algorithm comprising the Particle Swarm Optimization (PSO), Baum-Welch (BW), and Genetic Algorithms (GA) is proposed and implemented for optimizing the parameters of HMMs. The algorithm not only overcomes the shortcomings of the slow convergence speed of the PSO but also helps the BW escape from local optimal solution whilst improving the performance of GA despite the increase in the search space. Detailed experimental results demonstrates the effectiveness of our proposed approach when compared to other techniques available in literature.

*Corresponding author: E-mail: azizdanaa@gmail.com, azizdanaa@tatu.edu.gh;

Keywords: Hidden Markov Models; training; hybrid; genetic algorithm; particle swarm optimization.

1. INTRODUCTION

Hidden Markov Models (HMMs) are Machine Learning Algorithms which are used to model processes with a finite set of hidden/internal states governed by a set of transition probabilities. In a particular state, a visible observation symbol is emitted also according to an associated probability distribution [1]. The application of HMMs ranges from speech and image recognition, intrusion/anomaly detection in data, motion/action analysis in videos, to bioinformatics among others.

Rabiner, [1] also outlined the following general characteristics of a Hidden Markov Model;

1. The number of Hidden states denoted by N and represented as $S=S_1, S_2...S_N$ where $S_i = 1, 2, \dots, N$ are individual states denoted by q_t at a specific time t .
2. The number of unique observation symbols denoted by M , and usually specified by a set of symbols $V = V_1; V_2; \dots, V_M$, where $V_i, i = 1, 2, \dots, M$
3. A transition probability among states denoted by a matrix, $A = [a_{ij}]$ as defined in (1) and (2) below:

$$a_{ij} = P(q_{t+1} = S_j | q_t = S_i), 1 \leq i, j \leq N, t = 1, 2, \dots, T - 1 \quad (1)$$

Also,

$$\sum_{j=1}^N a_{ij} = 1, \text{ where } 1 \leq i \leq N. \quad (2)$$

1. An emission probability matrix, $B = [b_j(k)]$ as defined in (3) and (4) where;

$$B_j(w) = P(V_k = q_t | S_j = q_t) \quad (3)$$

where $1 \leq j \leq N, 1 \leq M$

Also,

$$\sum_{w=1}^M b_j(w) = 1, \text{ and } 1 \leq j \leq N \quad (4)$$

2. An initial probability for each state denoted by the vector $\pi = [\pi_i]$ as defined in (5) below;

$$\pi_i = P(q_1 = S_i), 1 \leq i \leq N, \quad (5)$$

such that $\sum_{i=1}^N \pi_i = 1$

HMMs should be able to address the following three (3) basic problems in order to make them applicable in solving real life problems [1]:

1. The Evaluation Problem: Given a sequence of observations, O and a Hidden Markov Model, λ , the task is to compute a probability for the observation sequence, $P(O|\lambda)$ with respect to the model.
2. The Decoding Problem: Determining the optimal state transition sequence for an underlying Markov process.
3. The Learning Problem: Given a series of observation, O , the task is to estimate the model parameters in order to maximize the probability of an observation sequence, $P(O|\lambda)$.

The solution to Problems 1 and 3 is what this study focusses on. The use of HMMs have become increasingly popular in the last several years due to the fact that, the models are very rich in mathematical structure and hence can form the theoretical basis for use in a wide range of applications. An effective optimization of the parameters of these Models for enhanced performance has remained computationally challenging and there is no generally agreed method that can guarantee best performance within reasonable computing time [2]. Various algorithms such as the Baum-Welch (BW), Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Tabu-Search (TS) among others have been proposed in literature for optimizing the parameters of HMMs [3].

The Baum-Welch (BW) algorithm is very popular estimation method due to its reliability and efficiency. However, it is easily trapped in local optimum and very sensitive to preliminary estimates [4]. GAs searches parallel from a population of points with the ability of avoiding being trapped in local optimal solution. When the number of possible solutions (chromosomes) which are uncovered to the genetic operators and mutation is large however, there is most likely to be an exponential increase in the search space which leads to a poor performance of the algorithm [5]. According to [6], PSO algorithm has emerged as a new training algorithm for Hidden Markov Models based on its simplicity and robust optimization capacity requiring small number of parameters and correspondingly lower number of iterations but has a relatively slow convergence speed.

A hybrid algorithm inspired by the BW, GA and PSO algorithms is proposed and implemented for optimizing the parameters of HMMs. The proposed algorithm overcomes the shortcomings of the slow convergence speed of the PSO and also enable the BW escape from local optimal solution whilst improving the performance of the GA by reducing its search space.

A possible drawback of not using this improved hybrid optimization algorithm will be less computationally efficient HMMs with poor low detection rates that may also suffer from slow convergence.

The rest of the paper is organized as follows: In Section 2, we present a review of related works. The methodology adopted for the study highlighting the working principle of an HMM and how our proposed hybrid algorithm is created is outlined in Section 3. Detailed experimental results and discussion to establish the efficacy of the proposed model is presented in Section 4. Finally, we conclude the paper with some discussions in Section 5.

2. RELATED WORKS

The theoretical aspects of Hidden Markov Models was carefully and methodically reviewed demonstrated by [1] on how they have been applied to selected problems in machine recognition of speech. He then applied them in speech recognition where the BW algorithm is used in training the proposed models. Experimental results established that, BW algorithm is very sensitive to initial estimates and can easily trapped in local optimal solution.

[7] Implemented GA in optimizing the parameters of HMMs to model speech signals where the model is encoded as chromosomes comprising two parts formed by concatenating the rows of the Transition and Emission probability Matrices. Although the Proposed algorithm has a lower convergence rate, it performed better than the BW procedure proposed by [1] in terms of recognition rate. A training method proposed by [2] based on GA and BW algorithms to optimize the parameters of obtain an optimized number of states in HMMs and other model parameters. Experiments with the 100 words extracted from the TIMIT corpus revealed that, although they overcame the slow convergence rate of a simple GA approach, their proposed models required more training time.

Tabu search (TS), an AI technique capable of searching for global optimal solution was proposed by [8] for training HMMs applied to speaker independent (SI) continuous speech recognition. However, when initial parameters and factors are not carefully chosen, TS do not show a great improvement over BW and GA as proposed by [2].

A Variable Population-size Genetic Algorithm (VPGA) was proposed by [3]. They introduced a "dying probability" for the individuals and subsequently proposed a PSO-GA-based hybrid algorithm (PGHA) based on the VPGA. The proposed algorithms converges faster than the PSO and GA to obtain a global solution. However, more training time is required to train the models to obtain optimum performance.

[9] proposed an improved training algorithm where the BW algorithm is applied to the new positions of particles in PSO to locally improve their positions. The BW algorithm is executed on the newly discovered positions of particles so that these positions will be locally improved. Each Model typically corresponds to a specific word unit although the training data for the speech usually consist of utterances and where the points separating the various segments of speech corresponding to each underlying sub-word model in the sequence is not exactly known. Experimental results revealed that the hybrid algorithm is superior to the BW algorithm and that proposed by [3] in terms of recognition rate. A hybrid algorithm consisting GA, TS and BW algorithms for HMM parameter optimization was implemented by [10]. Even though the proposed algorithm requires more training time, it does not only overcome the shortcoming of the slow convergence speed of GA and TS algorithms but also assists the BW algorithm escape from local optimum.

A framework for Hidden Markov Model training mainly based on the principles of utilizing Particle Swarm Optimization (PSO) concepts was also proposed by [11] where generating new states and updating likelihood values were the new components they included related to training HMM. [12] Proposed an optimized HMM with PSO algorithm named PSOHMM aimed at finding global optimal solutions. A re-normalization and re-mapping mechanisms to handle the constraints in HMM is developed and experiments revealed that PSOHMM can search better than BWHMM, with faster convergence

speed but performs poorly when the number of observation symbols is large.

3. METHODOLOGY

3.1 The Baum-Welch Algorithm

The Baum-Welch algorithm estimates the elements of the transition, emission and initial probability matrixes denoted as A, B and π respectively but may raise numerical exceptions and errors on computers when numbers computed by the algorithm are very small [5].

Consider an Observation sequence, $O = (O_0, O_1, O_2, \dots, O_{T-1})$ and a possible state sequence $X = (X_0, X_1, X_2, \dots, X_{T-1})$ where the interest is to compute the probability of the Observation sequence with respect to a given Hidden Markov Model λ . The Emission Probability Distribution Matrix (B) is formulated as in (6);

$$P(O|X, \lambda) = b_{X_0}(O_0)b_{X_1}(O_1) \dots b_{X_{T-1}}(O_{T-1}) \quad (6)$$

The initial probability distribution matrix(π) and state transition distribution matrix (A) may also be formulated as in (7) below;

$$P(X|\lambda) = \pi_{X_0}a_{X_0,X_1} \dots a_{X_{T-2},X_{T-1}} \quad (7)$$

(8) is obtained from (7)

$$P(O, X|\lambda) = P(O|X, \lambda)P(X|\lambda) \quad (8)$$

By summing over all possible state sequences, (9) is obtained

$$P(O|X) = \sum_X P(O, X|\lambda) \quad (9)$$

$$= \sum_X P(O|X, \lambda)P(X|\lambda) \quad (10)$$

Substituting (6) and (7) into (10), (11) is obtained

$$= \sum_X \pi_{X_0} b_{X_0}(O_0) a_{X_0,X_1} \dots a_{X_{T-2},X_{T-1}} b_{X_{T-1}}(O_{T-1}) \quad (11)$$

Using (11) is computationally intensive and so the forward algorithm to estimate $P(O|\lambda)$ with much more accuracy is proposed [1].

For $t = 0, 1, 2, \dots, T-1$ and $i = 0, 1, 2, \dots, N-1$, the probability of the partial observation sequence denoted as $\alpha_t(i)$, where the system is in state i at time t is defined in (12).

$$\alpha_t(i) = P(O_1, O_2, O_3, \dots, O_t, X_t = q_i | \lambda) \quad (12)$$

$\alpha_t(i)$ is calculated recursively as follows:

1. Let $\alpha_0(i) = \pi_i b_i(O_0)$,

$$\text{For } i = 0, 1, 2, \dots, N-1 \quad (13)$$

2. For $i = 0, 1, 2, \dots, N-1$ and $t = 1, 2, \dots, T-1$, we compute $\alpha_t(i)$ as in (14)

$$\alpha_t(i) = \sum_{j=0}^{N-1} [\alpha_{t-1}(j) a_{ji}] b_i(O_t) \quad (14)$$

3. Incorporating (12) into (14), (15) is obtained

$$P(O|\lambda) = \sum_{i=0}^{N-1} \alpha_{T-1}(i) \quad (15)$$

The backward algorithm, or β -pass is adopted to determine the optimal state sequence for a given observation sequence [13];

For $t = 0, 1, 2, \dots, T-1$ and $i = 0, 1, 2, \dots, N-1$, $\beta_t(i)$ is defined as in (16);

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \dots, O_{T-1} | x_t = q_i, \lambda) \quad (16)$$

Then $\beta_t(i)$ can be computed recursively as in (17) to (19):

1. for $i = 0, 1, 2, \dots, N-1$, Let $\beta_{T-1}(i) = 1$ (17)

2. for $i = 0, 1, 2, \dots, N-1$

and $t = T-2, T-3, T-4, \dots, 0$, $\beta_t(i)$ is defined as in (18);

$$\beta_t(i) = \sum_{j=0}^{N-1} [\beta_{t+1}(j) a_{ij}] b_j(O_{t+1}) \quad (18)$$

3. For $t = 0, 1, 2, \dots, T-1$ and $i = 0, 1, 2, \dots, N-1$, $\gamma_t(i)$ is defined as in (19);

$$\gamma_t(i) = P(x_t = q_i | O, \lambda) \quad (19)$$

The variable, $\gamma_t(i)$ can be defined in terms of $\alpha_t(i)$ and $\beta_t(i)$ as in (20),

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} \quad (20)$$

For $i, j \in \{0, 1, 2, \dots, N-1\}$ and $t = 0, 1, 2, \dots, T-2$, the variable, $\gamma_t(i, j)$ is used to represent the probability of the system being in state i at time t and moving to state j at time step $t + 1$, formulated as in (21):

$$\gamma_t(i, j) = P(x_t = q_i, x_{t+1} = q_j | O, \lambda) \quad (21)$$

In terms of α , β , A and B , $\gamma_t(i, j)$ from (21) can be defined as in (22);

$$\gamma_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O | \lambda)} \quad (22)$$

For $t = 0, 1, \dots, T-2$, the relationship between $\gamma_t(i, j)$ and $\gamma_t(i)$ is represented as in (23);

$$\gamma_t(i) = \sum_{j=0}^{N-1} \gamma_t(i, j) \quad (23)$$

3.2 Particle Swarm Optimization (PSO)

Proposed to simulate behaviors of birds searching for food or the movement of fishes' shoal iteratively in order to optimize a numeric problem. The flocking population is known as a swarm and the individuals are called particles which evaluate their positions relative to a global fitness. During each iteration, the particles share the values of their best positions in order to utilize those memories to adjust their next positions and velocities [14].

For a swarm consisting of n particles and an objective function, f , defined on a search space S , the i^{th} particle represents a D -dimensional vector X_i ($i = 1, 2, \dots, m$). It means that the i^{th} particle locates at $X_i(t) = (x_{i1}, x_{i2}, \dots, x_{iD})$ where $i = 1, 2, \dots, m$ in the searching space. The position of each particle is a potential result. The velocity of the i^{th} particle is also a D -dimensional vector, denoted as $V_i(t) = (v_{i1}, v_{i2}, \dots, v_{iD})$ where $i = 1, 2, \dots, m$.

The best solution the particle discovers since the start of the search is denoted as $X^+_{i}(t)$ and the best position of the entire swarm (social knowledge) denoted as $\hat{x}_i(t)$. During each iteration, the position and velocity of a particle, i , is computed as (24) and (25) respectively;

$$X_i(t+1) = X_i(t) + V_i(t) \quad (24)$$

$$V_i(t+1) = wV_i(t) + r_1 U([0,1]) (X^+_{i}(t) - X_i(t)) + r_2 U([0,1]) (\hat{x}_i(t) - X_i(t)) \quad (25)$$

$U([0,1])$ is a uniform random value chosen between 0 and 1 and the parameters w , r_1 and r_2 controls the entire particle system. The variable w is an inertia weight controlling the previous velocity of a particle. It is relatively easier to integrate PSO with other algorithms due to ability to run parallel computation. It is also robust and includes an intelligent search with a very good global optimization as compared to other metaheuristic algorithms. Furthermore, PSO rapidly converges to good solutions and has guaranteed convergence. On average, it performs efficiently when applied to comparatively complex problems [15].

3.3 The Proposed Hybrid Optimization Algorithm

Specifically, the Genetic algorithm which acts a bridge between the Baum-Welch and Particle Swarm Optimization algorithms is outlined as follows;

1. At $t=1$, an initial population $N(t)$ of 100 individuals (chromosomes) from using the Baum-Welch algorithm where each individual corresponds to a Hidden Markov Model is generated and their fitness values calculated.
2. $t=t+1$
3. While($t \leq 100$)

The next population is generated by keeping only the best 50 individuals of $N(t)$ by:

- i. Selecting randomly two individuals (λ_a, λ_b) from the population
 - ii. Recombining them with a multiple point crossover operator to obtain λ_c
 - iii. λ_c is mutated into λ'_c
 - iv. $N(t+1) = N(t) \cup \lambda'_c$
4. Let $N'(t)$ be the best 100 individuals found by the Genetic algorithm
 5. The PSO algorithm is applied to $N'(t)$

A linear recombination crossover that computes offsprings as a weighted sum of two parent models λ_a and λ_b , was adopted since the sum of each row in the Transition probability Matrix, observation Emission Matrix and the Initial Transition matrix equals 1.

Given two rows of the matrix, $R_1 = (X_1, X_2, \dots, X_N)$ and $R_2 = (Y_1, Y_2, \dots, Y_N)$, a new row, R_3 , is

computed using the linear recombination crossover as in (29).

$$R_3 = \alpha R_1 + (1-\alpha) R_2 = (\alpha X_1 + (1-\alpha) Y_1, \alpha X_2 + (1-\alpha) Y_2, \dots, \alpha X_N + (1-\alpha) Y_N) \quad (29)$$

The variable α denoted by αX is set to 0.5 for all crossovers so that the sum of the coefficients in each new row equals 1 as required. The value of

$P(O|\lambda)$ which represents the probability that an observation sequence is indeed generated by the Hidden Markov Model, λ , is an appropriate criterion adopted in this study to determine the fitness of a particular solution.

The details of the various electronic banking platforms were considered the internal states of the proposed model.

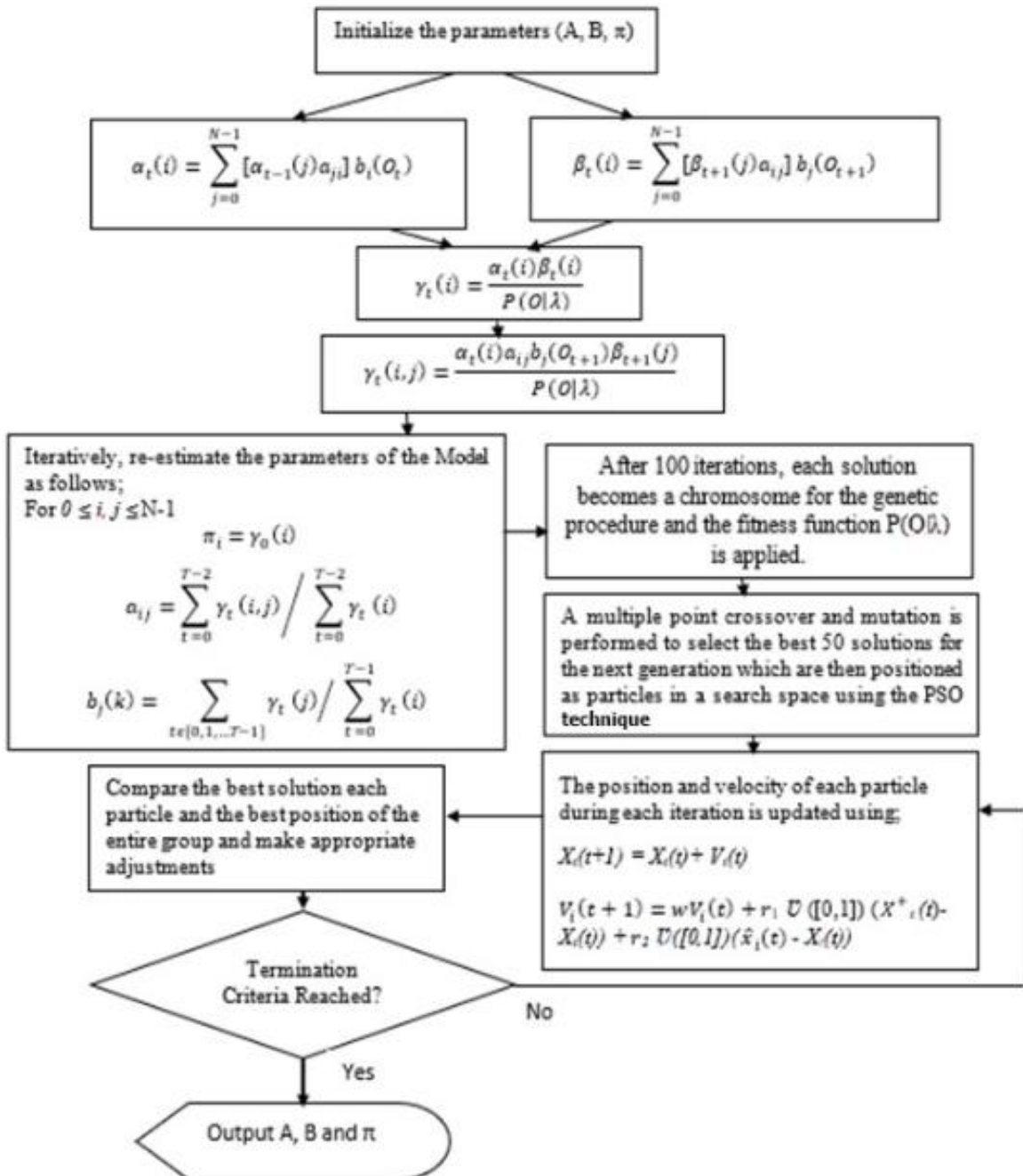


Fig. 1. The proposed hybrid optimization algorithm for HMMs

Transaction amounts are categorized as Low (l) = (0; 500], Medium (m) = (501; 1000], and High (h) = (1001; Transaction Limit] values. Also, the frequency at which they occur are also categorized into a Low (Less than 5 times a month), Medium (Between 5 and 10 times a month), and High (at least 10 times a month) are also considered by our proposed model. For example, if an accountholder performs about seven (7) transactions with the month with an average value of say 700, then the corresponding observation symbol is medium-frequency medium-amount (mm).

The transaction amounts are then combined with the frequency at which they occur in order to group customers according to their transaction profiles using the K-Means clustering algorithm. This results in the formulation of nine (9) observation symbols for the proposed model. The profiles considered are detailed in Table 1;

The combination (x, y, z) as in amount parameter refers to an accountholder who has to performed x percent of transactions in the high amount, y percent in medium amount, and z percent in the low amount categories.

On the other hand, (x, y, z) as in Frequency refers to an accountholder who has been found to carry out x, and z percent of his/her transactions on the low, medium and high frequencies respectively. The focus at this point is to establish how the system performs with different mixes of transaction amount ranges and frequency of transactions.

4. RESULTS AND DISCUSSION

To establish the efficacy of the proposed hybrid algorithm, it is used in optimizing the parameters of HMMs to detect anomalies in Electronic banking transactions and the results compared with employing the standard BW, GA, PSO algorithms and hybrids of any two of them.

Synthetic dataset generated using a simulator called BankSim which simulates electronic banking transactions based on a sample of real transactions extracted from financial logs from a bank was employed [16].

After the training phase, series of observations, O, is constructed from an account holders training data corresponding to the various transaction profiles as outlined in Table 1 and the algorithms used to evaluate the probability, $P(O/\lambda)$ of generating O with the model.

For each number of hidden state (N), fifty (50) simulation runs were performed with each of the algorithms and the average value of P (O/ λ) computed to enable us compare their performances using the same set of emission symbols as outlined in Section 3.

For a low-amount, low-frequency transaction profile, the values of P(O/ λ) and their averages for the different values of N is also shown in Fig. 2 and Table 2 respectively.

Table 1. Transaction mix representing the various transaction profiles of customers

Transaction Profile	Parameter	Transaction mix
High Amount, High Frequency	Amount	(60,30,10)
	Frequency	(70,25,5)
High Amount, Medium Frequency	Amount	(80,10,10)
	Frequency	(25,75,5)
High Amount, Low Frequency	Amount	(90,7,3)
	Frequency	(25,5,75)
Medium Amount ,High Frequency	Amount	(30,60,10)
	Frequency	(80,15,5)
Medium Amount ,Medium Frequency	Amount	(15,75,10)
	Frequency	(15,80,5)
Medium Amount ,Low Frequency	Amount	(10,70,20)
	Frequency	(15,5,80)
Low Amount, Low Frequency	Amount	(3,2,90)
	Frequency	(30,20,50)
Low Amount ,Medium Frequency	Amount	(5,10,85)
	Frequency	(30,50,20)
Low Amount ,High Frequency	Amount	(10,10,80)
	Frequency	(50,30,20)

Table 2. Average values of P (O/λ) for a low-amount, Low-frequency transaction profile for all values of N

Algorithm	Number of states (N)				Average
	2	3	4	5	
BW	0.689	0.731	0.723	0.521	0.666
GA	0.685	0.712	0.785	0.742	0.731
PSO	0.754	0.751	0.684	0.631	0.705
BWPSO	0.775	0.785	0.802	0.769	0.783
BWGA	0.762	0.795	0.811	0.768	0.784
GAPSO	0.785	0.892	0.821	0.803	0.825
Proposed	0.824	0.932	0.852	0.832	0.860

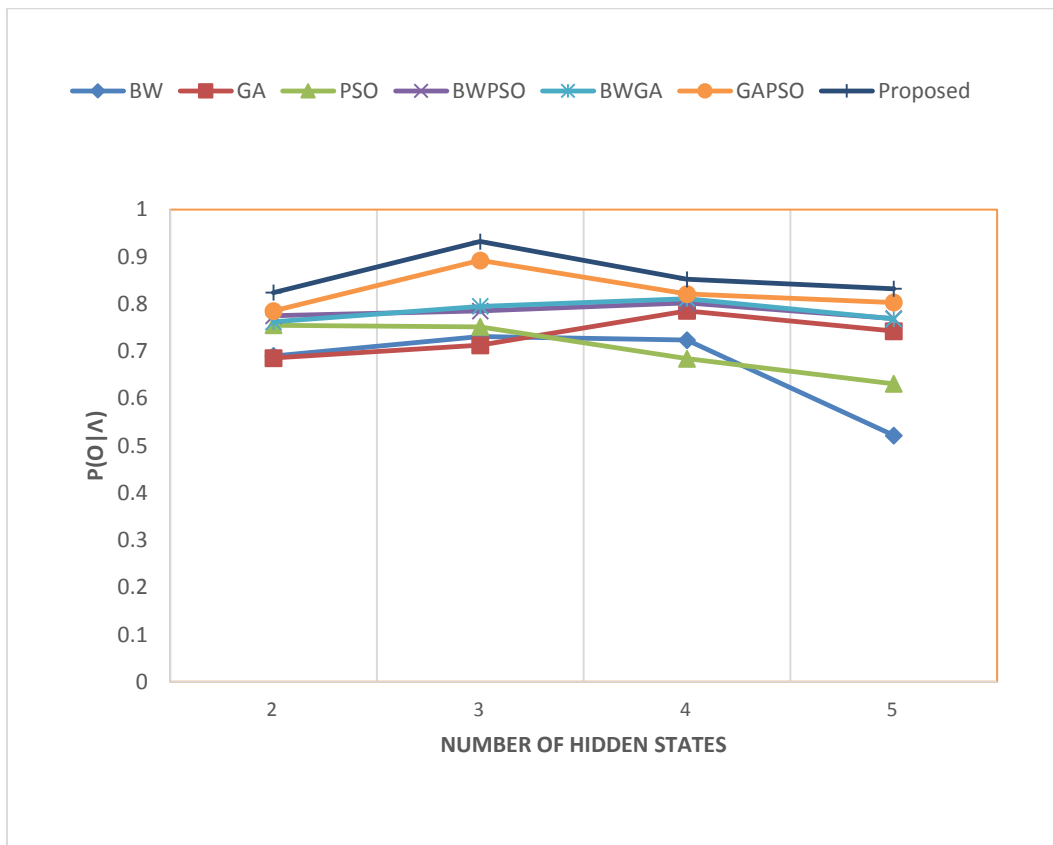


Fig. 2. Values of P (O/λ) for a low-amount, low-frequency transaction profile for all values of N

Comparatively, Fig. 2 reveals that, for all the different values of N, the BW algorithm produces the worst results whilst our proposed hybrid algorithm produces the best. PSO and GA produce average values which are still below 80%. The results for all the nine (9) observation sequences representing the different transaction profiles of customers are displayed in Fig. 3.

It's evident from Fig. 3 that, for all transaction profiles, optimizing HMMs with our proposed

algorithm results in better detection rates for all possible values of N.

The efficiency of our proposed algorithm in terms of the time taken to optimize the HMMs for all the transaction profiles is shown in Fig. 4.

Fig. 4 reveals that, on the average, BW algorithm recorded relatively better training times although it is trapped in local optimum as can be seen in the relatively lower detection rates (P (O/λ)) in Fig. 3. It can also be seen that, the PSO algorithm has a relatively slow convergence

speed. Our proposed hybrid approach recorded lower execution times just like the BW algorithm but performs better in terms of recognition rates

for all the possible number of hidden states as shown in Fig. 3.

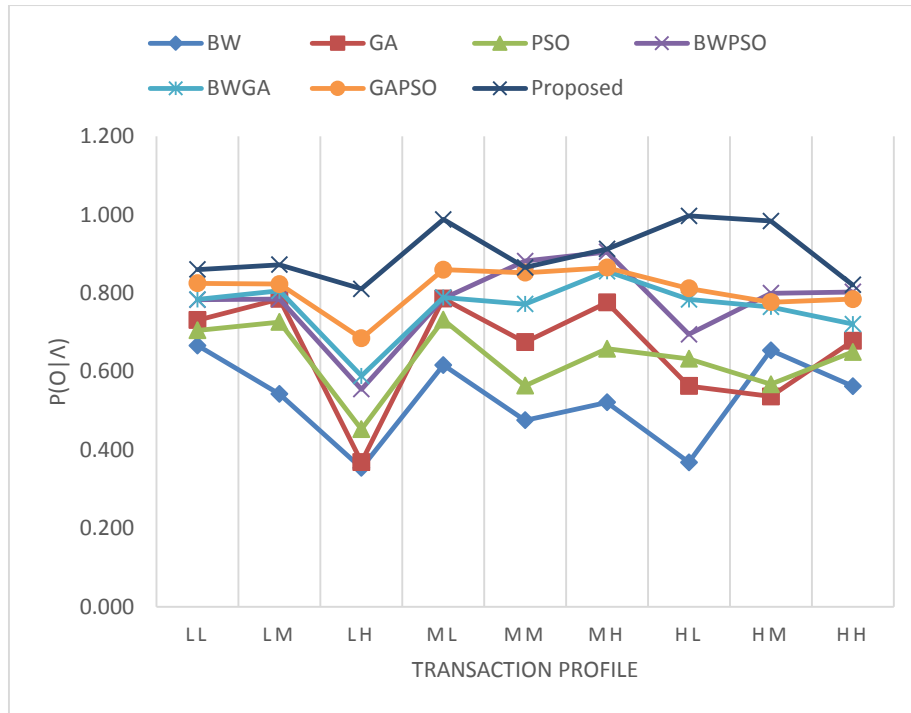


Fig. 3. Values of $P(O|\lambda)$ for the various transaction profiles for the various hidden number of states

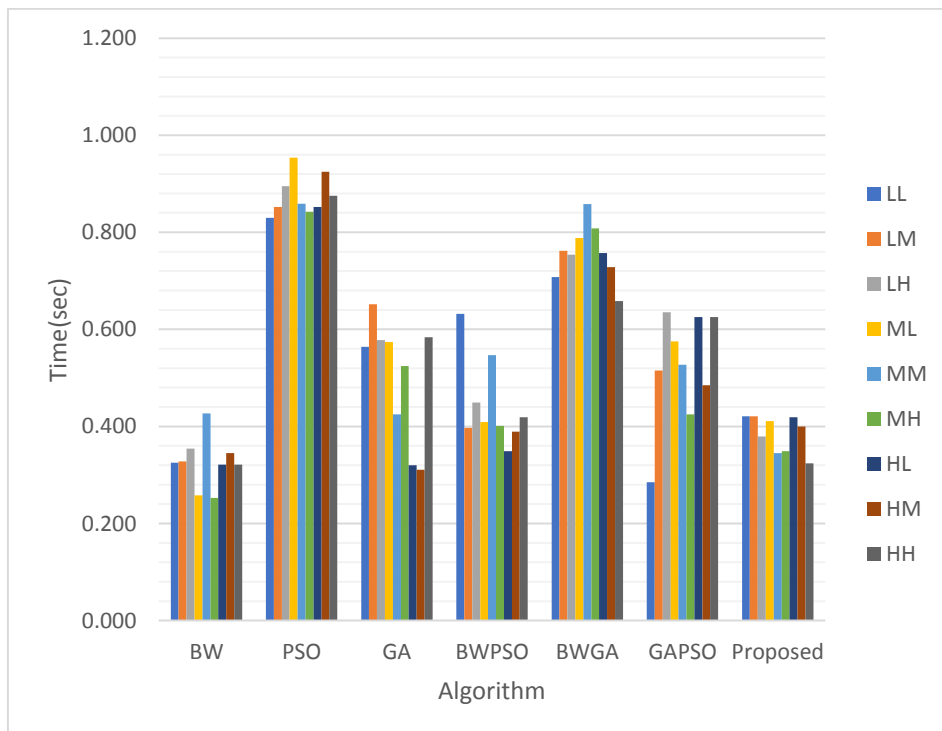


Fig. 4. Time taken to optimize the HMMs by the various algorithm for all transaction profiles

5. CONCLUSION

A hybrid optimization algorithm which leverages on advantages of the Baum-welch, Genetic and Particle Swarm Optimization algorithms have been proposed and implemented in optimizing the parameters of Hidden Markov Models for improved performance. The proposed algorithm overcomes the shortcomings of the slow convergence speed of the PSO and also enable the BW escape from local optimal solution whilst improving the performance of the GA by reducing its search space.

The Baum-Welch algorithm estimates reasonably rather than random guess the particles of the Genetic algorithm. However, when the number of elements which are exposed to mutation is large in Genetic algorithm, there is often an exponential increase in search space size which leads to a poor performance. The PSO algorithm which has guaranteed convergence and averagely performs efficiently when applied to comparatively complex problems is then introduced to produce a global solution of the model parameters. The execution time of the proposed algorithm even though slightly higher than that obtained by the BW and the PSO algorithms, it still produced better recognition rates. The results are promising and future work will focus on applying the proposed hybrid algorithm to optimize the parameters of HMMs in the area of speech and image recognition, fraud detection in data, motion/action analysis in videos, bioinformatics among others.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. LR Rabiner. A tutorial on hidden markov models and selected applications in speech recognition, *proc. IEEE*; 1989. DOI: 10.1109/5.18626
2. Kwong S, Chau CW, Man KF, Tang KS. Optimisation of HMM topology and its model parameters by genetic algorithms, *Pattern Recognit*; 2001. DOI: 10.1016/S0031-3203(99)00226-5
3. Shi XH, Liang YC, Lee HP, Lu C, Wang LM. An improved GA and a novel PSO-GA-based hybrid algorithm, *Inf. Process. Lett.* 2005;93(5):255–261. DOI: 10.1016/j.ipl.2004.11.003
4. Chang L, Ouzrout Y, Nongaillard A, Bouras A. Optimized Hidden Markov Model based on Constrained Particle Swarm Optimization; 2018.
5. Aupetit S, Monmarché N, Slimane M. Hidden Markov models training by a particle swarm optimization algorithm, *J. Math. Model. Algorithms.* 2007;6(2):175–193. DOI: 10.1007/s10852-005-9037-7
6. Haikuan L, Dachao Y, Lei Z, Zhiyuan L, Dawei J. A new improved simplified particle swarm optimization algorithm, *J. Phys. Conf. Ser.* 2019;1187(4). DOI: 10.1088/1742-6596/1187/4/042077
7. Chau CW, Kwong S, Diu CK, Fahrner WR. Optimization of HMM by a genetic algorithm, *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.* 1997;3:1727–1730. DOI: 10.1109/icassp.1997.598857
8. Thatphithakkul N, Kanokphara S. HMM parameter optimization using Tabu search, *IEEE Int. Symp. Commun. Inf. Technol. Isc.* 2004;2:904–908. DOI: 10.1109/iscit.2004.1413850
9. Fengqin Y, Changhai Z. An effective hybrid optimization algorithm for HMM, in *Proceedings - 4th International Conference on Natural Computation, ICNC*; 2008. DOI: 10.1109/ICNC.2008.367
10. Yang F, Zhang C, Bai G. A novel genetic algorithm based on tabu search for HMM optimization,” *Proc. - 4th Int. Conf. Nat. Comput. ICNC.* 2008;4(1):57–61. DOI: 10.1109/ICNC.2008.365
11. Hewahi NM. Particle swarm optimization for hidden markov model, *Int. J. Knowl. Syst. Sci.* 2015;6(2):1–15. DOI: 10.4018/ijkss.2015040101
12. Chang L, Ouzrout Y, Nongaillard A, Bouras A. Optimized Hidden Markov Model based on Constrained Particle Swarm Optimization. 2018;02(2):1–5.
13. Stamp M, Stamp M. A revealing introduction to hidden markov models, *Introd. to Mach. Learn. with Appl. Inf. Secur.* 2018;7–35. DOI: 10.1201/9781315213262-2
14. Hassan R, Cohanim B, De Weck O, Venter G. A comparison of particle swarm optimization and the genetic algorithm, *Collect. Tech. Pap. - AIAA/ASME/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf.* 2005;2:1138–1150. DOI: 10.2514/6.2005-1897

15. Ning J, Zhang Q, Zhang C, Zhang B. A best-path-updating information-guided ant colony optimization algorithm, Inf. Sci. (Ny). 2018;433–434:142–162. DOI: 10.1016/j.ins.2017.12.047
16. Lopez-Rojas EA, Axelsson S. Banksim: A bank payments simulator for fraud detection research, in 26th European Modeling and Simulation Symposium, EMSS 2014; 2014.

© 2021 Danaa et al.; This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Peer-review history:

*The peer review history for this paper can be accessed here:
<http://www.sdiarticle4.com/review-history/69810>*