

## Computational solution of networks versus cluster grouping for social network contact recommender system

Arnold Adimabua Ojugo, Debby Oghenevwe Otake

Department of Mathematics / Computer Science, Federal University of Petroleum Resources Effurun, Nigeria

### Article Info

#### Article history:

Received Jan 26, 2020

Revised Feb 20, 2020

Accepted Mar 13, 2020

#### Keywords:

Cluster  
Ego-centric networks  
Implicit contact  
Recommender  
Social graphs  
Tie-strength

### ABSTRACT

Graphs have become the dominant life-form of many tasks as they advance a structure to represent many tasks and the corresponding relations. A powerful role of networks/graphs is to bridge local facts that exist in vertices as they blossom into patterns that help explain how nodal relations and their edges impacts a complex effect that ripple via a graph. User cluster are formed as a result of interactions between entities. Many users can hardly categorize their contact into groups today such as “family”, “friends”, “colleagues” etc. Thus, the need to analyze such user social graph via implicit clusters, enables the dynamism in contact management. Study seeks to implement this dynamism via a comparative study of deep neural network and friend suggest algorithm. We analyze a user’s implicit social graph and seek to automatically create custom contact groups using metrics that classify such contacts based on a user’s affinity to contacts. Experimental results demonstrate the importance of both the implicit group relationships and the interaction-based affinity in suggesting friends.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



### Corresponding Author:

Arnold Adimabua Ojugo,  
Department of Mathematics / Computer Science,  
Federal University of Petroleum Resources Effurun,  
Delta State, Nigeria.  
Email: [ojugo.arnold@fupre.edu.ng](mailto:ojugo.arnold@fupre.edu.ng)

## 1. INTRODUCTION

Group communication has brought about more effective interaction that is usually validated amongst a cluster of people. It seeks to broadcast data over a communication channel and/or medium to a cluster of people – restricting peer-to-peer communications. Email, text and social platforms allow the support for group conversations and consequently, sharing of data in formats like photo, links and document [1]. Despite the prevalence of such group communication, users spend less time creating and maintaining custom contact group. Social platforms provide users with exclusive relationship and links with their corresponding contacts. Thus, it is common practice that some users can identify a person as a friend – even when they do not know him/her. Treating all contacts in same manner has been the basis for fraud and other social engineering vices. There is the need for users to differentiate and classify into groups, their personal contacts as this is quite a safe and unrestrictive practice. Allowing users to curb and minimize the fears associated with contacts data sharing and interaction on a social networks. Many users have had to quit groups when close relations (family, friends and colleagues) are added or removed from the platform [2].

Many cluster relationships are easily modeled as social graph that are implemented via social networks. A graph is a symbolic representation of a network and of its connectivity as a structure of linked nodes and their relationship. Mathematically, a graph is a set of vertex (node)  $V$ , connected by edges  $E$  and denoted as  $G = (V, E)$ . A social graph interconnects users showing their relationships on a social network

especially in relation to a user's egocentric personal social graph/network [3]. It denotes a discrete graph containing vertices, linked together by edges that describes nodal relationship between these entities [4, 5]. Thus, it describes a socially weighted network that analyses relationship or ties between users, entities or objects by means of their interaction as they share data.

**Literature Review: Types of Social Graphs:** There are two kinds of social graph namely in the light of social discuss namely: explicit and implicit graph relationship. The implicit social graph describes an interaction between users, their contacts and group of contacts. It defines a graph whose vertices are not represented as explicit data objects in memory; But rather, are determined algorithmically from some more concise inputs. It is also a graph whose edges are weighed by feats such as frequency, recency, and direction of interaction between users and their contacts and their group of contacts. They are used to identify clusters of contacts who form groups that are meaningful and useful to each user [1, 6].

Conversely, explicit graphs are such that two-individuals deliberately and mutually describe their connection with one another. Thus, such graphs can be mined more easily, since they begin with hard data, and not algorithms that will be hard for competitors to replicate in the future. It is best understood as truly personal and social [7]. Thus, explicit graphs are rare and examples include Facebook and LinkedIn. Thus, groups change dynamically as new users are added to multi-party communication threads; while, others are also removed. Thus, a person's individual relationship dynamically evolve and changes over time as a friend becomes one's family, a colleague becomes a friend, a friend becomes a colleague etc. The need to consistently update all relationships users have with their contacts, require constant maintenance, which is tedious and time consuming feats [2].

**Navigating Many Clustering Models:** A graph attempts to bridge local features, as they blossoms into global patterns, to explain how nodal relationships impacts a complex effect that ripple through a population system. Each node shapes a graph's evolution as need arises. Social graphs seeks two goals: (a) better understand how networks evolve, and (b) study the dependent social processes like innovation diffusion and data retrieval via models to specify how local interaction of nodal feats are explored to a global pattern [8]. Graph binds together nodes via a predefined model so that we can effectively analyze its entities along theories that sought to explain the inherent observed patterns [9-10]. Thus, they propagate local feats present in the nodes that eventually emerge as global patterns. It examine dynamics in relationship between nodes as well as helps to locate all the influential entities within such a network – as it theoretically, allows connection convergence of nodes [5].

## 2. MATERIALS AND METHODS

### 2.1. Statement of problem

The following problems are to be addressed:

- a. Manual creation of groups from user contacts is quite time consuming as the user must deliberately identify clusters from his/her contact list so as to create the required groups.
- b. The dynamic nature of social groups especially with the addition, deletion and amending of relationships etc, users often manually handle such updates of custom groups.
- c. Study seeks to tackle the above problem using an implicit social graph through a friend suggest algorithm.

The study seeks to implement a predicted suggestive algorithm using the implicit social graph. This will help automatically help users to create custom contact groups in their phonebook. It will further, intelligently help eliminate time penalty and cost associated and spent to manually create custom contact graph. The study specific goals include: (a) to describe an interaction-based metric for estimating user's affinity to his contact group, (b) implement the deep neural network and friend suggestion algorithm via a user's implicit social graph, (c) compare results of generated groups using predefined labels (seed-set dataset) of contacts as already categorized as friends by the user, (d) to suggest contacts that can be used to expand seed-set of group, (e) to demonstrate the importance of implicit group relationship and the interaction-based affinity in suggesting contacts to add/remove from groups, and (f) to compare both models and their effectiveness in these classifications.

### 2.2. Data gathering

Study employs the Enron Email Corpus Dataset that consist of a large collection of employee's email messages of the Enron Corporation collected during the legal investigation of Enron accounting fraud in December 2001. It contains over 600,000 messages from 150-users; But, for the study, we consider just a single user (i.e. an employee) email – because we seek a user' egocentric network. Each employee folder in the dataset, has other folders of emails such as incoming message and outgoing message folder. The employee email address to be considered is Shackleton Sara (chosen based on a balance in the number of

incoming and outgoing messages through a thorough check on all employees' email folders). Folders considered in Shackleton Sara folders are inbox folder, notes\_inbox folder, sent folder and sent item folder.

The rationale for the adoption of the dataset used, is based on (a) standard email for social networks, (b) the dataset explores benefits and tie-strength of users, and (c) dataset incorporates the various components metrics to be measured for the graph algorithm. Dataset is obtained from [web]: <http://www.cs.cmu.edu>.

### 3. EXPERIMENTAL FRAMEWORK

Machine learning seeks to develop models, algorithms and systems that mimics intelligence as it allows such systems to evolve its behavior based on empirical data from sensors data and databases. The system explores data mining methodology, tools and techniques to capture characteristics of interest as we seek its underlying unknown probability distribution. Thus, illustrating the relations between observed and historic data. Thus, system seeks to automatically learn to recognize complex patterns and make intelligent decisions from it [11-14].

#### 3.1. Friend suggest algorithm (FSA)

Following the work of Roth et al (2010), the researcher further explores the friend suggest algorithm which probes the presence of implicit clustering in a user's egocentric network by observing groups of contacts who are frequently present as co – recipients in the same email threads. FSA functions within the egocentric network in order to show suggestions based only on a user's local data so as to protect user privacy and avoid exposing connections between the user's contacts that may not otherwise have been identified to him. The inputs to friends suggest is a seed, which is a small set of one or more contacts that belong to a group. This seed may be characterized by the user picking a few contacts. e.g. As an initial list in the "To" field of an email. Given this seed, FSA finds the contacts in the user's egocentric network that are related to the seed (i.e. they are present in the same implicit clusters). FSA further returns a score for each suggested contact, indicating the goodness of its fit to the existing seed-set. The FSA is applicable to problems of group clustering any interaction based social graph [1, 6]. Figure 1 describes the various components that makes up the architecture of the friend suggest algorithm (FSA).

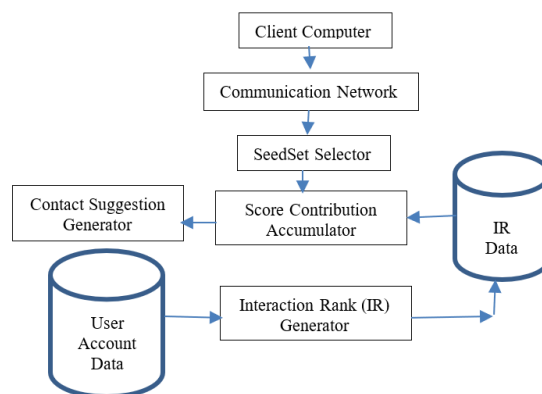


Figure 1. General architecture of the friend suggest algorithm

FSA consist of these components explained below as in [6]:

- a. The Interaction Rank – Here, the implicit social graph has that its weighed edges represent tie-strength relationship between a user and his implicit group. This is computed via the following criteria: (i) Frequency – These are groups with which a user interacts occasionally are more important to the user than groups with which he interacts infrequently, (ii) Recency is ability for group to change and be dynamic over time, (iii) Direction: Interactions a user initiates is of more significance, and (iv) Contact importance: Groups that includes important contact are also more important to a user than other groups. Thus, we adjust the edge weights based on a contact-importance metric, a group that includes one or more important contacts are assigned a greater edge weight that group that do not include such important contact. The contact-importance metric for a respective contact of a user may be determined by other data about user's relationship or global information about the contact's position within the socio-centric graph. Furthermore, receiving an email from a contact (passive interaction) is a weaker signal of closeness than an active interaction of sending an email to that contact.

- b. The Core Routine Function expands (adds/removes contact) via the seedset in the ego-centric network.
- c. The Scoring Function – implements the various versions of the Update\_Score algorithm, and consists of:
  - (i) intersecting group score, (ii) intersecting weighted score, (iii) intersect group count, (iv) top\_contact score, and (v) suggesting contact to remove algorithms.

Finally, extending [1], study seeks to explore the FSA taking advantage of its strength on a custom dataset.

### 3.2. Spectral deep learning network

Deep neural network has successfully been implemented in systems that seek to learn useful features and construct multi-layer networks from a vast amount of training data. Forecast accuracy is improved using DNNs, allowing more data about a raw dataset to be obtained. DNN deep architectures including multiple hidden layers—and each hidden layer alone conducts a non-linear transformation from the previous layer to the next [15, 16]. With deep learning proposed by [17], DNN is trained according to two sections: (a) pre-trained, and (b) fine-tuned procedures [18, 19].

Auto-Encoder: [20] Auto-Encoder is a type of unsupervised three-layered neural network whose output target is an input data shown in Figure 2. It includes both encoder and decoder networks. The encoder network transforms input data from a high-dimensional space into a low-dimensional space; While, the decoder network remodels the input from the previous step. The encoder network is defined as encoding function  $f_{\text{encoder}}$  – with encoding process as in (1) – where  $x^m$  is a data point and  $h^m$  is the encoding vector obtained from  $x^m$ :

$$h^m = f_{\text{encoder}}(X^m) \quad (1)$$

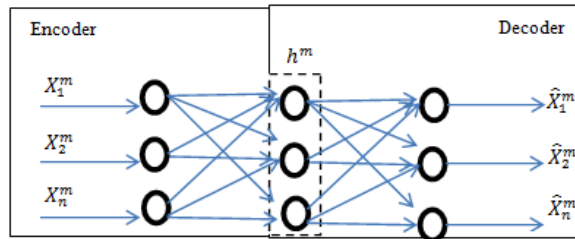


Figure 2. Auto-encoder and decoder in deep neural network

The Decoder is a reconstruction function denoted as  $f_{\text{decoder}}$  as in (2), where  $x^m$  is the decoding vector obtained from  $h^m$ . Other algorithms for encoding (reconstruction function) includes (3-5) respectively.

$$X^m = f_{\text{decoder}}(h^m) \quad (2)$$

$$\text{logsig: } f_{\text{encoder}}(X^m) = \frac{1}{1 + e^{-X^m}} \quad (3)$$

$$\text{satline: } f_{\text{encoder}}(X^m) = \begin{cases} 0 & \text{if } X^m \leq 0 \\ z & \text{if } X^m < 1 \\ 1 & \text{if } X^m \geq 1 \end{cases} \quad (4)$$

$$\text{pureline: } f_{\text{encoder}}(X^m) = X^m \quad (5)$$

Pre-Training: N auto-encoders can be stacked to pre-train N-hidden-layer DNN. When given an input dataset, the input layer and the first hidden layer of the DNN are treated as the encoder network of the first auto-encoder. Next, the first auto-encoder is trained by minimizing its reconstruction error. The trained parameter set of the encoder network is used to initialize the first hidden layer of the DNN. The first and second hidden layers of the DNN are regarded as the encoder network of the second auto-encoder. Accordingly, the second hidden layer of the DNN is initialized by the second trained auto-encoder. This continues until the  $N_{\text{th}}$  auto-encoder is trained to initialize the final hidden layer of the DNN [20]. Thus, all hidden layers of the DNN are stacked in an auto-encoder in each training N times, and are regarded as pre-trained. This pre-training process is proven to be significantly better than random initialization of DNN and quite useful in achieving generalization in many of the classification tasks [21, 17].

Fine-Tuning is a supervised process that improves performance of a DNN. The network is retrained, training data are labelled, and errors calculated by difference between real and predicted values are back-propagated using stochastic gradient descent (SGD) for all multi-layer networks. SGD randomly selects data samples, and iteratively updates gradient direction with weight parameters. Best gradient direction is obtained with a minimum loss function. The merit of SGD is that it converges faster and also does not consider the entire dataset. Making it far suitable for complex neural networks [20] as in (6) below:

$$E = \frac{1}{2} \sum M(y_i - t_i)^2 \quad (6)$$

E is loss function, y is real label and t is net output. Gradient of weight w is obtained as derivative of error equation – so that with the gradient of the weight  $w_{ij}$ , the updated SGD equation is defined by (7) [20]:

$$W_{ij}^{\text{new}} = W_{ij}^{\text{old}} - h(y_j - t_j) * y_i(1 - y_i)\omega \quad (7)$$

$h$  is the step-size and it is greater than 0; while,  $\omega$  is number of hidden layers in DNN [20]. The process is optimized and tuned using the weights and threshold of the correctly labelled data. This enables the DNN to learn important knowledge for its final output and direct the parameters of entire network to perform correct classifications [14].

### 3.3. K-nearest neighbourhood (KNN)

Is a well-known supervised learning model for pattern recognition, introduced by Fix and Hodges in 1951. It remains one of the most popular nonparametric models for classification problems [22]. KNN assumes that observations, that are close together, are likely to have the same classification. The probability that a point  $x$  belongs to a class is estimated by proportion of training points in a specified neighbourhood of  $x$  that belong to that class. This point(s) is then either classified by majority vote or by a similarity degree sum of the specified number ( $k$ ) of nearest points. In majority voting, number of points in neighbourhood belonging to each class is counted, and the class to which the highest proportion belongs to is most likely classification of  $x$  [13]. Similarity degree sum calculates a similarity score for each class based on the K-nearest points and classifies  $x$  into the class with the highest similarity score. Its lower sensitivity to outliers allows majority voting to be used other than similarity degree sum [23]. We use majority voting for data to determine which points belongs to neighbourhood so that distances from  $x$  to all points in the training set must be calculated. Any distance function that specifies which of two points is closer to the sample point could be used. The most common distance metric used in K-nearest neighbour is Euclidean distance [24] given by Eq. 8 as distance between each test point  $f_t$  and training set point  $f_s$ , each with  $n$  attributes as thus:

$$d = [(f_{t1} - f_{s1})^2 + (f_{t2} - f_{s2})^2 \dots + (f_{tn} - f_{sn})^2]^{\frac{1}{2}} \quad (8)$$

In general, KNN performs the following steps: (a) chosen of  $k$  value, (b) distance calculation, (c) distance sort in ascending order, (d) finding  $k$  class values, and (e) finding the dominant class [25]. A challenge with KNN is to determine the optimal size of  $k$  that acts as smoothing parameter. A small  $k$  is not sufficient to accurately estimate population proportions around the test point. A larger  $k$  will result in less variance in probability estimates (but for risk of introducing more bias).  $K$  should be large enough to minimize probability of a non-Bayes decision, and small enough that all points included, gives an accurate estimate of the true class. [26] in [13] found optimal value of  $k$  to depend on sample size and covariance structures in each population, and on proportions for each population in the total sample. In cases where the differences in covariance matrices and the difference between sample proportions are either both small (or both large), then optimal  $k$  is  $N^{3/8}$  ( $N$  is number of samples in training set). If there is a large difference between covariance matrices, and a small difference between sample proportions (or vice-versa), optimal  $k$  is determined by  $N^{2/8}$ .

This model presents some merits [27]: (a) mathematical simplicity allows for achieve classification results as good or even better than other more complex pattern recognition techniques, (b) it is free of statistical assumptions, (c) its effectiveness does not depend on the space distribution of classes, and (d) when the boundaries between classes are not hyper-linear or hyper-conic, K-nearest neighbour performs better than LDA [13].

Its major demerits is that it does not work well for large differences in samples in each class. KNN yields poor data about the structure of its classes, and relative importance of variables in classification. Also, it does not allow graphical representation of the results, and in case of large number of samples, computation become excessively slow. In addition, KNN requires more processing and memory needs than other methods. All

prototypes in training set must be stored in memory and used to calculate Euclidean distance from every test sample. The computational complexity grows exponentially as the number of prototypes increases [28, 13].

### 3.4. Linear discriminant analysis

LDA is an effective supervised classification method with wide range of applications. Its theory is to classify compounds (rules) dividing n-dimensional descriptor space into two regions separated by a hyper-plane that is defined by linear discriminant function. Discriminant analysis transforms classification tasks into functions with data partitioned into classes. It reduces the problem to an identification of a function [13]. The focus of discriminant analysis is to determine this functional form (assumed to be linear) and estimate its coefficients. Introduced in 1936 by Ronald Fisher, LDA seeks the mean of a set of attributes for each class, and using the mean of these means as boundary. It thus projects attribute points onto the vector that maximally separates their class means, and minimizes their within-class variance as in (9) [29]:

$$LDA = X'(X_2 - X_1) - \frac{1}{2}(X_2 + X_1)S^{-1}(X_2 - X_1) > c \quad (9)$$

X is vector of the observed values,  $X_i$  ( $i = 1, 2, \dots$ ) is the mean of values for each group, S is sample covariance matrix of all variables, and c is cost function. If the misclassification cost of each group is considered equal, then  $c = 0$ . A member is classified into one group if the result of the equation is greater than c (or = 0), and into the other if it less than c (or = 0). A result that equals c (set to 0) indicates such a sample cannot be classified into either class, based on the features used by the analysis. LDA function distinguishes between two classes – if a data set has more than two classes, the process must be broken down into multiple two-class problems. The LDA function is found for each class versus all samples that were not of that class (one-versus-all). Final class membership for each sample is determined by LDA function that produced the highest value and is optimal when variables are normally distributed with equal covariance matrices. In this case, the LDA function is in same direction as Bayes optimal classifier [30], and it performs well on moderate dataset in comparison to other more complex method [13]. Its mathematical function is simple and requires nothing more complicated than matrix arithmetic. It assumes linearity in the class boundary, however, limits the scope of application for linear discriminant analysis. When boundaries are nonlinear, the performance of the linear discriminant may be inferior to other classification methods. Thus, to curb this – we adopt a decimal encoding of the data to give us a semblance of linear, continuous boundaries [31].

### 3.5. Rationale for choice of algorithms

Stochastic models are inspired by biological populations and laws of evolution. They search for optimal solution via hill-climbing method that is flexible, adaptive to changing states and suited for real-time tasks. They guarantee high convergence in multimodal task(s) via an initialized random population and allocates increasing trials to regions found with high fitness for optimality in time. Its demerit is in their inefficiency with linear systems in that if the optimal is in small region surrounded by regions of low fitness – function becomes difficult to optimize. But, the adoption of such stochastic graph-based models entails that for iterated hill-climbing – once a peak is located, it restarts with another, randomly chosen, starting point. Its merit is simplicity and with each random trial performed in isolation, no overall picture of the domain is obtained. As the evolution progresses, it continues to allocate its trials evenly over a search space. This means that the algorithm continues to evaluate as many points in the regions found to be of low fitness as in regions found to be of high fitness [11-12].

## 4. RESULT, DISCUSSION AND FINDINGS

### 4.1. Model performance

Performance is evaluated using the mean square error (MSE), mean regression error (MRE), mean absolute error (MAE) and coefficient of efficiency (COE) as in Table 1 [13]. It shows that the adopted models for DNN has a mean square error of .73, mean regression error of 0.79, mean absolute error of 0.75 and coefficient of efficiency of 0.581 respectively. Also, we have that the FSA has a mean square error of .41, mean regression error of 0.51, mean absolute error of 0.45 and coefficient of efficiency of 0.781 respectively. Then LDA has a mean square error of .18, mean regression error of 0.21, mean absolute error of 0.43 and coefficient of efficiency of 0.492 respectively. Lastly, KNN has a mean square error of .36, mean regression error of 0.21, mean absolute error of 0.23 and coefficient of efficiency of 0.853 respectively. It is worthy of note that the more the values of MSE, MRE and MAE tends to 1, the more accurate the adopted model's prediction; While, the value of COE conversely should tend more to zero (0).

Table 1. Model convergence performance evaluation

<i>Model</i>	<i>MSE</i>	<i>MRE</i>	<i>MAE</i>	<i>COE</i>
DNN	.73	.79	.75	.581
FSA	.41	.51	.45	.781
LDA	.18	.21	.43	.492
KNN	.36	.31	.23	.853

To ensure model's efficiency and accuracy, we compute misclassification rate and improvement percentages for both training and testing dataset, summarized in Table 2 and Table 3 respectively via (10, 11)

$$MR(A) = \frac{\text{No of Incorrect Classify}}{\text{No of Sample Set}} \quad (10)$$

Table 2 shows model's error for misclassification of the emails. KNN, LDA and FSA yields error of 30.3%, 39.7% and 18.3%; while, DNN yields 12.8% respectively.

$$\text{Improvement} = \frac{MR(A) - MR(B)}{MR(A)} * 100 \quad (11)$$

Table 3 shows LDA, KNN and FSA yields an improvement of 41.1%, 43% and 69.9% respectively; while, DNN yields 76%. It was also observed that though KNN is quite sensitive to the relative magnitude of different attributes, all attributes are thus scaled by their z-scores before using KNN [32].

Table 2. Misclassification rate of each model

<i>Model</i>	Classification Errors	
	Training Data	Testing Data
KNN	31.2%	30.3%
LDA	43.4%	39.7%
DNN	13.5%	12.8%
FSA	19.3%	18.3%

Table 3. Improvement percentage rate of each model

<i>Model</i>	Improvement Rate %	
	Training Data	Testing Data
LDA	45.83%	41.16%
KNN	41.79%	43.09%
DNN	78.78%	76.33%
FSA	69.30%	69.91%

#### 4.2. Classification accuracy

The Figure 3 shows the prediction accuracy of the various models with FSA showing an accuracy of 92-percent, DNN showing an accuracy of 89-percent, KNN showing an accuracy of 74-percent and LDA showing an accuracy of 70-percent respectively. It is also worthy of note that in order forms of classification, DNN (deep neural networks) have been found to outperform these models.

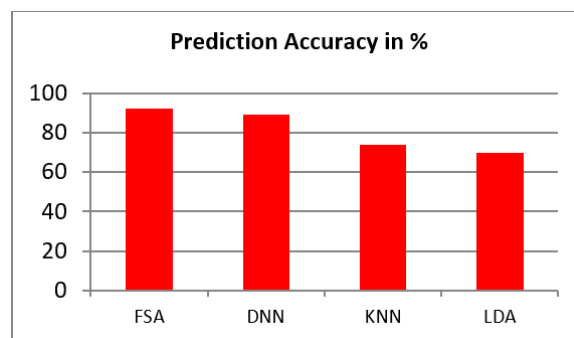


Figure 3. Prediction Accuracy of Algorithms in percentage

#### 4.3. Processing speed

The Figure 4 shows the mean prediction processing time for the various models with FSA showing an men processing time of 1.22, DNN has mean processing time of 0.98, KNN shows a mean processing time of 2.98 and LDA shows a mean processing time of 3.36 respectively. It becomes clearer thus, that DNN reach an accuracy of 89-percent at 0.98 seconds, FSA reached an accuracy of 92-percent at 1.22 seconds. KNN reached an accuracy of 74-percent at 2.98seconds; while, LDA reached its 70-percent at 3.36 seconds respectively.

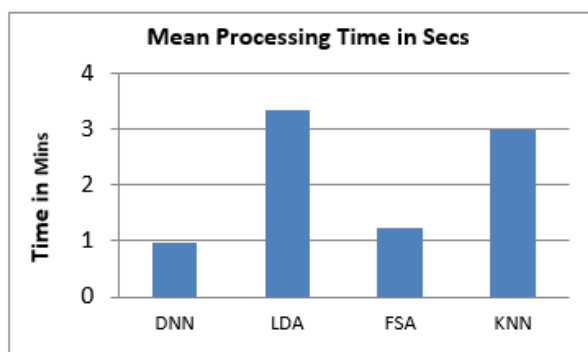


Figure 4. Processing time in seconds

#### 4.4. Convergence time

The *rationale* for model choice(s) is to compare and measure their convergence behavior amongst other statistics as seen in Figure 3 – 5 respectively. Also, LDA and KNN converged after 405 and 387-iterations respectively; While, FSA and DNN converged after 253- and 193-iterations respectively. DNN outperforms FSA for the task being considered. We *note*, model's speed is traded-off for greater accuracy of classification, a greater number of rule set generated to update the knowledge database for optimality and greater functionality.

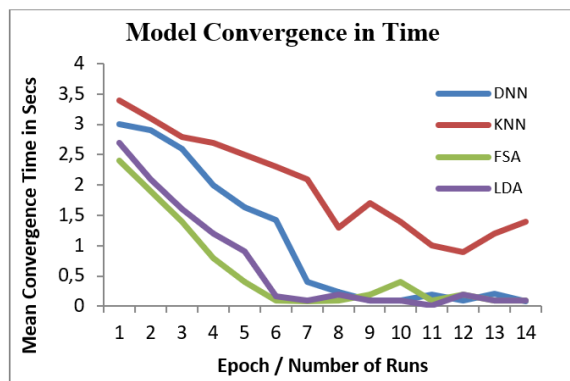


Figure 5. Convergence time of matches

## 5. CONCLUSION

There has to be effective communication between two entities for an effective relationship. For this to happen, both persons have to be in constant communication. A means through which this happens, is via social platforms such as link editing, photo sharing, and email communication and so on. All models used shows that two or more users can connect using an implicit social graph. They all generate a friend's group, given a small seed-set of contacts already categorized by a user, as friends, colleagues groups. They equally suggest contacts to expand the seed-set as contained in the user's egocentric network. For the study, we used the Enron Corpus Email dataset to determine the interaction rank (edge weight) between a user and his group



of contacts. We detailed the comparative statistics of the various models based on the efficiency, speed, convergence and others.

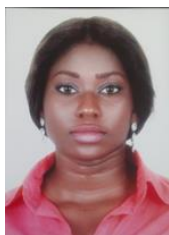
## REFERENCES

- [1] Roth, M., Barenhoiz, T., Ben-David, A., Deutscher, D., Flysher, G., Hassidim, A., Horn, I., Leichtberg, A., Leiser, N., Matias, Y., Meron, R., "Suggesting friend through friends suggest algorithm using implicit social graph," *Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2010.
- [2] Raad, E., Chbeir, R., Dipanda, A., "Discovering relationship types between users using profiles and shared photos on a social network," *Multimedia Tools and Applications*, pp. 1-30, 2011.
- [3] Gilbert, E and Karahalos, K., "Predicting tie strengths with social media," *J. Computer and Human Interface*, ACM 978-1-60558-246-7/09/04, vo. 15, pp. 76-97, 2009.
- [4] Pathapati, S.K., Venugopalan, S., Kumar, A.P., Bhamidipaty, A., "People and entity retrieval in implicit social networks," *2011 IEEE 5th International Conference on Internet Multimedia Systems Architecture and Application*, 2011
- [5] Ojugo, A. A., Aghware, F., Yoro, R.E., Yerokun, M.O., Eboka, A.O., Anujeonye, C.N., Efozia, F., "Predicting behavioral evolution diffusion on a graph model," *Advances in Networks*, vol. 3, no. 2, pp. 8-21, 2015.
- [6] Ojugo, A. A., Otakore, O. D., "Intelligent clustering connectionist friendship recommender system using implicit graph estimation for social networks," *Technical Report of the Federal University of Petroleum Resources Effurun (TR-FUPRE-2391)*, 2015.
- [7] Sherman, M., "implicit vs explicit social graph," *Clipperhouse*, 2011, [Online], Available: <https://clipperhouse.com/implicit-vs-explicit-social-graphs/>, [Accessed on aug 4, 2018]
- [8] Ojugo, A. A., F. Aghware., R. E. Yoro., M. O. Yerokun., A. Eboka., C. Anujeonye., F. Efozia., "Predicting behavioral evolution on a graph-based model," *Advances in Networks*, vol. 3, no. 2, pp. 8-21, 2015.
- [9] Granovetter, M., "Economic action and social structure: The problem of embeddedness," *American J. Sociology*, vol. 91, no. 3, pp. 481–510, 1985.
- [10] Acemoglu, D., Carvalho, V.M., Ozdaglar, A., Tahbaz-Salehi, A., "The network origins of aggregate fluctuations," *Econometrica*, vol. 80, no. 5, pp. 96, 2012.
- [11] Ojugo, A.A., A.O. Eboka. "Modeling the computational solution of market basket associative rule mining approaches using deep neural network," *Digital Technologies*, vol. 3, no. 1, pp. 1–8, 2018.
- [12] Ojugo, A.A., A.O. Eboka., "Comparative evaluation for performance adaptive model for spam phishing detection," *Digital Technologies*, vol. 3, no. 1, pp. 9–15, 2018.
- [13] Ojugo, A.A., D.O. Otakore., "Improving early detection of gestational diabetes via intelligent classification models: a case of the Niger Delta region of Nigeria," *Journal of Computer Science & Application*, vol. 6, no. 2, pp. 82-90, 2018.
- [14] rate of hepatitis-B virus infection in Niger Delta region of Nigeria using graph-based diffusion heuristic model," *IJCAOnline International Journal of Computer Application*, vol. 179, no. 39, pp 27–33, 2018.
- [15] Schmidhuber, J. "Deep learning in neural networks: An overview," *Neural Netw*, vol. 61, pp. 85–117, 2015.
- [16] Bengio, Y.; Courville, A.; Vincent, P. "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, pp. 1798–1828, 2013.
- [17] Hinton, G.E.; Salakhutdinov, R. R., "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, pp. 504–507, 2006.
- [18] Zheng, Y.; Liu, Q.; Chen, E.; Ge, Y.; Zhao, J. L. "Time series classification using multi-channels deep convolutional neural networks," in *Web-Age Information Management*; Springer: Berlin, Germany, pp. 298–310, 2014.
- [19] Glorot, X.; Bengio, Y. "Understanding the difficulty of training deep feedforward neural networks," *In Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS'10)*, Sardinia, Italy, pp. 249–256, 13–15 May 2010.
- [20] Ma, T., Wang, F., Cheng, J., Yu, Y., Chen, X., "A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks," *Sensors*, vol. 16, no. 10, pp. 1701, 2016.
- [21] Erhan, D.; Bengio, Y.; Courville, A.; Manzagol, P.A.; Vincent, P.; Bengio, S., "Why does unsupervised pre-training help deep learning?" *J. Mach. Learn. Res.* vol. 11, pp. 625–660, 2010.
- [22] Fix, E. and Hodges, J., "Discriminatory analysis -Nonparametric discrimination: Consistency properties," *Project No. 2-49-004, Report No. 4, Contract No. AF 41(128)-31*, USAF School of Aviation, Randolph Field, Texas, 1951.
- [23] Chaovalitwongse, W., "On the time series k-nearest neighbor classification of abnormal brain activity," *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, vol. 37, no. 6, pp. 1005 - 1016, 2007.
- [24] Viaene, S., Derrig, R., Baesens, B., and Dadene, G., "A comparison of state-of-the art classification techniques for expert automobile insurance claim fraud detection," *The Journal of Risk and Insurance*, vol. 69, pp. 373-421, 2002
- [25] Yildiz, T., Yildirim, S., Altılar, D., "Spam filtering with parallelized KNN algorithm," *Akademik Bilisim*, 2008
- [26] Enas, G. and Choi, S., "Choice of the smoothing parameter and efficiency of k-nearest neighbor," *Computers and Mathematics with Applications*, vol. 12, no. 2, pp. 235-244, 1986.
- [27] Berrueta, L., Alonso-Salces, R., Heberger, K., "Supervised pattern recognition in food analysis," *Journal of Chromatography A*, vol. 1158, no. 1-2, pp. 196-214, 2007.
- [28] Muezzinoglu, M. and Zurada, J., "RBF-based neurodynamic nearest neighbor classification in real pattern space," *Pattern Recognition*, vol. 39, no. 5, pp. 747-760, 2006.
- [29] Vapnik, V., *Statistical learning theory*, New York: Wiley, 1998.
- [30] Billings, S. and Lee, K., "Nonlinear Fisher discriminant analysis using a minimum squared error cost function and the orthogonal least squares algorithm," *Neural Networks*, vol. 15, no. 2, pp. 262-270, 2002.
- [31] Duda, R., Hart, P., and Stork, D., *Pattern classification*, New York: John Wiley & Sons, Inc. 2001
- [32] Antal, P., Fannes, G., Timmerman, D., Moreau, Y., Moor, B.D., "Bayesian applications of belief networks and multilayer perceptrons for ovarian tumor classification with rejection," *Artificial Intelligence in Medicine*, vol. 29, no. 1-2, pp. 39-60, 2003.

## BIOGRAPHIES OF AUTHORS



Arnold Adimabua Ojugo received BSc in Computer Science from the Imo State University Owerri in 2000, MSc in Computer Science from the Nnamdi Azikiwe University Awka in 2005, and PhD in Computer Science from Ebonyi State University Abakiliki in 2013. He currently lectures with the Department of Mathematics/Computer Science of the Federal University of Petroleum Resources Effurun, Delta State, Nigeria. His research interests: Intelligent Systems and Control, High Performance and Dynamic Computing, Machine-Learning, Computer Vision, Ubiquitous Computing, Data Security and Forensics. He is an Editor with the Progress for Intelligent Computation and Application, Advancement for Scientific and Engineering Research and SciencePG (ACIS, AJNC, NET and WCMC Journals). He is a member of: The Nigerian Computer Society, Computer Professionals of Nigeria and International Association of Engineers (IAENG).



Debby Oghenevwe Otokore received BSc in Computer Science from Benson Idahosa University, Edo state in 2008, MSc in Computer Science from the University of Port-Harcourt 2015, She is currently pursuing a Ph.D in computer science at the University of Port-Harcourt, Nigeria. She currently lectures with the Department of Mathematics/Computer Science of the Federal University of Petroleum Resources Effurun, Delta State, Nigeria. Her research interests: Recommender and Intelligent Systems. She is a member of: The Nigerian Computer Society.