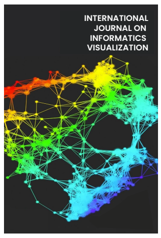
JOIV : Int. J. Inform. Visualization, 6(1-2): Data Visualization, Modeling, and Representation - May 2022 168-176

INTERNATIONAL JOURNAL
ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv



An Assessment Algorithm for Indoor Evacuation Model

Khyrina Airin Fariza Abu Samah [a,*], Amir Haikal Abdul Halim [a], Zaidah Ibrahim [b]

[a] *Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA Cawangan Melaka Kampus Jasin, Melaka, Malaysia*
[b] *Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA Shah Alam, Selangor, Malaysia*
Corresponding author: [*]khyrina783@uitm.edu.my

Abstract— **The public buildings increased significantly with the economy's growth and the population's advancement. The complexity of the indoor layout and the involvement of many people cause the indoor evacuation wayfinding to the nearest exit to be more challenging during emergencies such as fire. In order to overcome the problem, each building is compulsory to follow the standard evacuation preparedness required by Uniform Building By-Law (UBBL). Researchers have also developed evacuation models to help evacuees evacuate safely during the evacuation from a building. However, building owners do not know which evacuation model is suitable for implementing the chosen high-rise building. Two problems were identified in choosing a suitable evacuation model during the decision-making process. First, many developed evacuation models focus on studying different features of evacuation behavior and evacuation time. Second, the validation and comparison of the evacuation model is the missing process before applying the suitable evacuation model. Both validation and comparison procedures were made independently without any standard assessment that encapsulates the critical incident features during the indoor evacuation and virtual spatial elements. Therefore, this research proposed an indoor evacuation assessment algorithm to solve the problem. The assessment algorithm refers to the elements developed in our previous study. We determined attributes, executed simulations, and evaluated the cluster performance using the developed framework. The outcome can help the building owners assess which suitable existing evacuation model is the best to implement at the chosen building.**

Keywords— **Assessment algorithm; evacuation model; indoor evacuation; integrated assessment model; microscopic.**

Manuscript received 17 Jan. 2022; revised 20 Mar. 2022; accepted 14 Apr. 2022. Date of publication 31 May 2022.

I. INTRODUCTION

An evacuation is the retreat, dispersal, or withdrawal of individuals from locations of risk or threat and their reception and treatment in secure environs, all organized, regulated, and monitored [1]. The large public buildings increased significantly with the economy's growth and the population's advancement. Many people gathered in significant buildings, such as libraries, shopping centers, stages, and metro stations [2]. Therefore, it is essential to have the proper emergency evacuation in every high-rise building.

In an emergency, adequate contingency plans will facilitate egress and save lives. Evacuation is not the same as the pre-designed evacuation plan, considering different aspects, including the enclosed layout, environmental change, evacuees' features, and likely encounters with evacuees' crowds. Evacuees with a misperception of the building environment may display significant rounding or even be trapped, resulting in an exceptionally longer evacuation time. People generally follow a route of self-estimated rapid escape based on a sense of the current situation [1]. Furthermore, intense hysteria and stamping can contribute to multiple individuals evacuating in emergencies. The layout of escapes from structures, human psychology and behavior, and numerous social and behavioral patterns can heavily impact evacuation performance, leading to trapped [3]. Thus, it is a need for a high-rise building to have an evacuation model to allow evacuees to exit the building without any harm safely.

The evacuation model is divided into microscopic, macroscopic, and multiscale or mesoscopic models [4]. Microscopic models often define each individual's spatial and temporal actions [5]. The macroscopic model, also known as the continuum model, combines variables and monitors parameters such as bottleneck densities [6]. Lastly, individuals are considered but not individual interactions in mesoscopic models, focusing on groups but providing more precise information about each pedestrian. The idea is to maintain some control over the individual (for example, the permanence of a particular pedestrian in a specific region) while moving the group as a whole and avoiding local interactions [7]. The unit analysis is the individual; however, it does not consider the interaction between individuals [8]. Thus, mesoscopic is not considered in this research.

In evacuation assessments, microscopic and macroscopic models are commonly employed to depict pedestrian traffic [9]. Individuals are not generally described in macroscopic models, which only depict overall population movement. Microscopic models, on the other hand, concentrate on individual details. They have been utilized in several crowd simulation studies; to better understand crowd behavior in emergencies [10]; thus, microscopic models have been used widely in recent years [11]. Generally, researchers have mainly used these three models: Agent-Based Model (ABM), Social Force Model (SFM), and Cellular Automata (CA) for microscopic models [12]. Thus, the microscopic model is the best among the three types of evacuation models for the indoor evacuation model.

Therefore, this study proposed to develop an assessment algorithm to solve the two highlighted problems encountered when choosing the suitable evacuation model for the building. The research problem is: 1) many developed evacuation models focus on studying different evacuation behavior and evacuation time. Hence, the models differ in terms of the features, and 2) validation and comparison of the evacuation model is the missing process before applying the suitable evacuation model. Both procedures were made independently without any standard assessment that encapsulates the critical incident features during the indoor evacuation. We believed the proposed assessment algorithm was able to solve the problems. Fig. 1 shows the process involved in selecting the suitable indoor evacuation model. It starts with the evacuation preparedness, evacuation models, choosing suitable, and finally applying the chosen model. We found that one process is missing between the third and fourth process: assessing the suitable evacuation models. The process is crucial since not all buildings can use the same evacuation model.

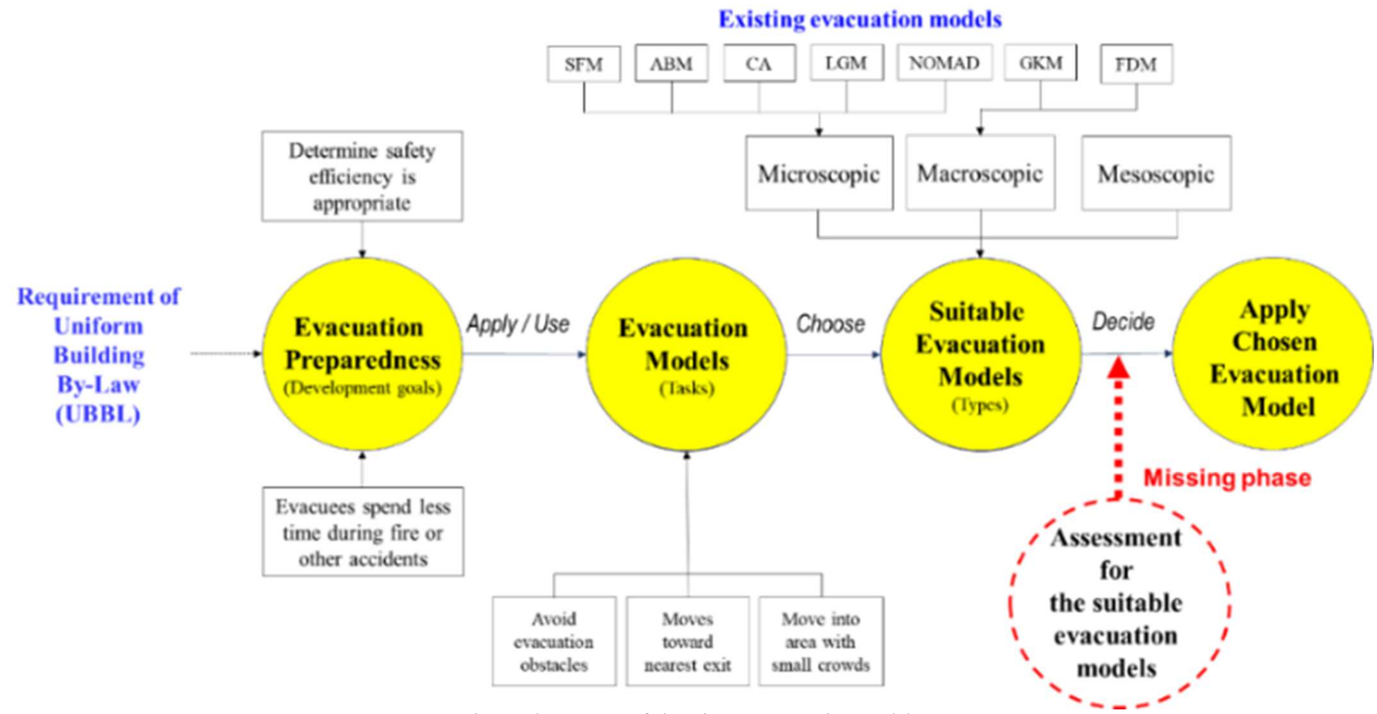


Fig. 1 The process of choosing an evacuation model

This research uses the help of an assessment model to develop the assessment algorithm. Integrated Assessment (IA) can be defined as knowledge regarding a problem domain that is integrated and available for societal learning and decision-making [13]. Integrated Assessment Model (IAM) is a method that combines information from two or more areas into a unified framework [14]. This research uses IAM to guide in creating the assessment algorithm. The IAM framework design is based on the ten dimensions of IAM. The dimensions are chosen based on the suitability of the research, and the framework's adaptation can then proceed. Fig. 2 shows the developed IAM framework for the indoor evacuation assessment algorithm [15]. The latest amendment is established on the recent research conducted. A total of eight chosen factors from the IAM dimensions are applied based on three outlier categories: 1) key drivers of integration, 2) methodological aspects requiring integration, and 3) aspects of the system to integrate.

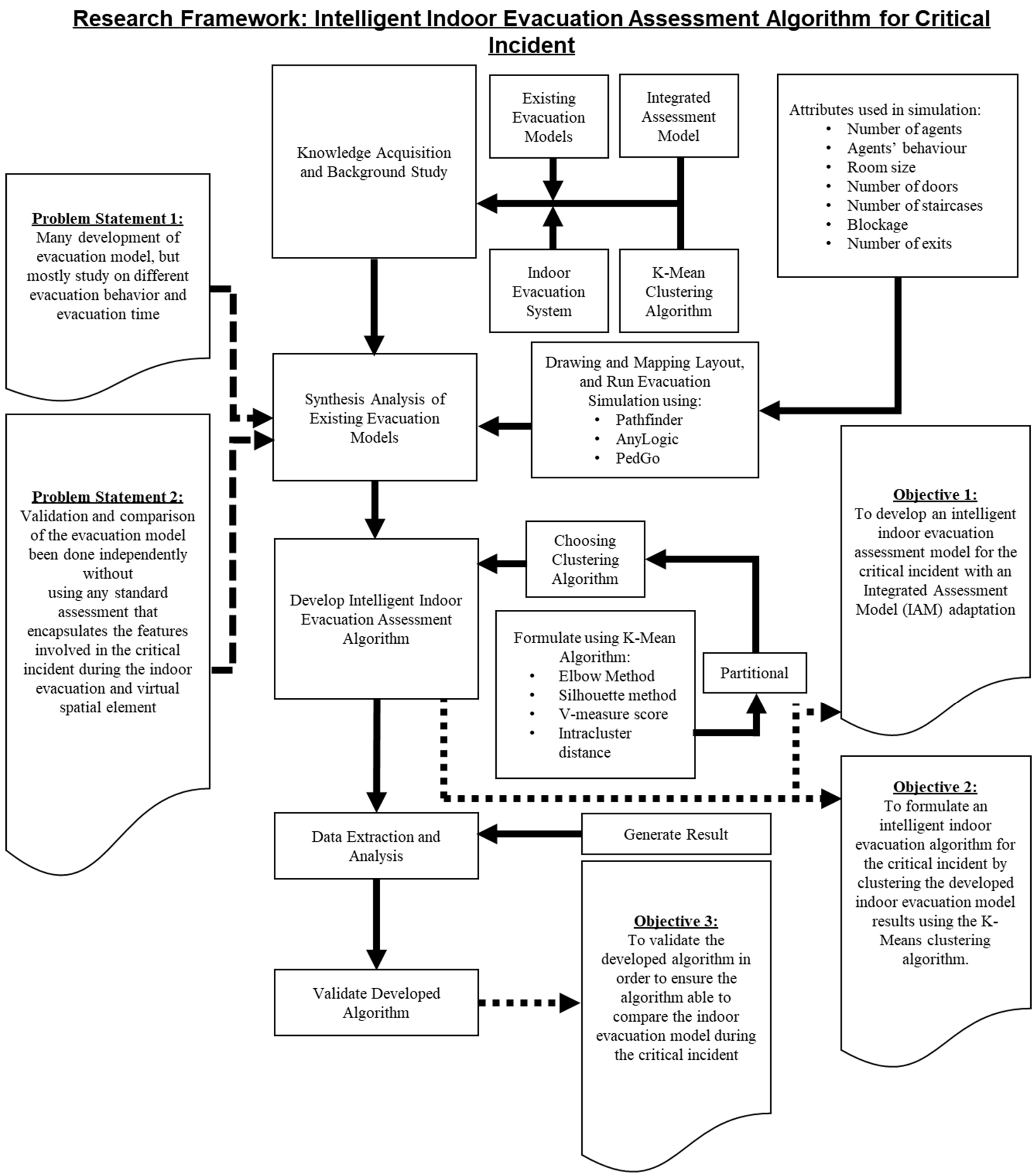


Fig. 2 IAM conceptual framework for indoor evacuation assessment algorithm

Based on the adapted IAM framework, five main steps are involved in creating the assessment algorithm. It starts with knowledge acquisition and background study until the developed algorithm is validated. Each step has its related subsets that help accomplish the step's goal. Two issues were identified during the second step:

- Many evacuation models were developed, but each focused on different evacuation behavior and time. Hence the models differ in their features and lead to users' hard decision to choose the best evacuation models.
- Each of the evacuation model's validation and comparison was conducted independently. There is no standard assessment that encapsulates the critical incident features during the indoor evacuation and virtual spatial element.

Thus, the first objective, to develop an intelligent indoor evacuation assessment algorithm for a critical incident with an IAM adaptation, is achieved during step three. This paper will further explain the implementation of the second objective: to formulate an intelligent indoor evacuation assessment algorithm for critical incidents by clustering the evacuation model results using the K-Mean clustering algorithm. The last objective is to validate the developed algorithm to ensure the algorithm can compare the indoor evacuation model during a critical incident. A custom validation questionnaire form and two experts evaluate the assessment algorithm.

The case study for this research is conducted on levels 13 and 14 floors of Yayasan Melaka's Building. Level 14th floor is the highest floor of the building. The simulation software used to gather time taken results is Pathfinder, AnyLogic, and PedGo. The simulation software represents the evacuation model chosen, ABM, SFM, and CA, respectively. The layout of levels 13 and 14 in a format of .dwg is implemented in each simulation software. Mapping and drawing of the specific levels based on the simulation studies created. A few fixed rules were followed during the mapping process. The rules are as follows:

- The pathways are fixed for each of the simulations.
- The agents are placed in the same room for each simulation software.

Such rules are followed because each simulation software is different in its functional capabilities. Thus, some simulations need to manually map the pathway for agents from starting point to the endpoint so that the agents can move during simulation. Agents distributed in the same rooms for each simulation software ensure fair simulations. For mapping, the steps are taken to map the layouts in each simulation software also differ from each other.

This paper's organization begins with a brief introduction and literature review in Section 1. Section 2 explains materials and for this research. Section 3 elaborates on the results and discussions. Section 4 concludes the study and briefly mentions future enhancement.

II. Materials and Methods

This section is divided into six sections for detailed design and development: 1) determine attributes, 2) execute the simulation, 3) find optimal *k* number, 4) evaluate clusters performances, 5) calculate intracluster distance, and 6) determine the best evacuation model.

A. Determine Attributes

The Seven attributes were determined as input for the simulation. The description of the attributes is presented in Table 1. For a detailed explanation, the number of agents plays an essential role in evacuation simulation. A study conducted uses various agents ranging between 10 to 30. The author concluded that the higher the number of agents evacuated, the higher the number of people who died. It was caused by the jam during the evacuation and also influenced by other factors, including blockage [16].

Next, individuals prefer to form groups with others when fleeing huge crowds. This group can assist in quickly locating an exit, although it could not always speed up the evacuation process. In crowds, behavioral features of people can lead to group orientation and exclusion, as well as convergence, conflict, balance, and imbalance of various energies [17]. These are why attribute behavior is used to study the time taken between grouped and scattered agents. Next, building infrastructure is essential for evacuation as it influences the time taken by agents to exit. Therefore, the attribute of room size must not be overlooked. For instance, a simulation conducted by Liu and Parhizgar [18] has different sizes of the room used to study and compare the evacuation time taken. The room size does play a role in affecting the time taken for agents to escape.

TABLE I
LIST OF ATTRIBUTES AND DESCRIPTION

No	Attributes	Description of the Purposes
1	Number of agents	The number of agents can affect the time taken to exit. Each experiment will use various agents to study the impact
2	Agents' behavior	The behavior is either grouped or scattered
3	Room size	To differentiate the effect of agents evacuating from a large or small room
4	Number of doors	A room possible to has many entries. Possibility if the doors are blocked or not functional such as being locked
5	Number of staircases	To determine if more or fewer staircases can slow down or speed up the evacuees evacuating the building
6	Number of blockages	It can happen at the agents' exit pathways in a real-life evacuation scenario. The blockage can be assumed as anything such as rubbles, obstacles, but mainly fire. Two blockages are placed near the exits for each simulation study that has the status of blockage is set as "yes"
7	Number of exits	There is always a possibility that the exits can be blocked or locked

The number of doors is considered as it can influence the time taken for agents to reach the exits. Conducted a simulation that produces a result where two doors always result in a faster evacuation time than one door [18]. If comparing a single or double door at the back, a double door can reduce evacuation time. Three doors do not provide any substantial advantages over two. Staircases are a common way to exit a high-rise building. At the same time, the design of the staircases, the agent's behavior, and different types of situations can influence the time taken for agents to escape a building [19]. Thus, this research will consider the number of stairs available to study their effect on the time taken for agents to escape.

Subsequently, the blockage is also an attribute for this assessment algorithm. A study by [20] claimed that the blockage could influence the time taken if it happened near the exit. However, some studies argue that the obstacle's beneficial effect might be noticed. Despite that, it was apparent that blockage in the room will disrupt the agents' evacuation path by consuming space or blocking the pedestrians' view if they reach a particular height [21]. It concluded that any blockages could affect agents' time to escape. Thus, the blockage is included to compare the time taken. Two blockages are placed near the exits for each simulation study with the status of blockage as "yes".

Next, the number of exits is an attribute because agents will generally need to exit the building. The number of availabilities of the exits can influence the time taken for agents to exit. A study determines the number of exits available, either one or three or five out of six exits. Even the number of agents is different for each simulation where it increases. It concluded that once the number of agents increases and the number of exits door decreases, the time to exit will increase [22]. Thus, this research determines how many exits are available in the simulation studies.

B. Execute Simulation

After drawing and mapping the layout and determining the value of the chosen attributes, simulation can be executed. Each of the simulations performed is based on the simulation studies created for the simulations. Each simulation's time taken result is essential since the different attributes' value can affect agents' evacuation time. The simulation software involved for each evacuation model chosen is Pathfinder for ABM, PedGo for CA, and AnyLogic for SFM.

C. Find Optimal k Number

In the scientific community, the k-means clustering method is regarded as one of the most powerful and widely used data mining techniques [23]. K mean can have a problem where it can get stuck in the wrong local optimal solutions. However, a good initialization has proven to be an excellent way to overcome this problem [24]. The clustering aims to improve the objective feature (*f*) by measuring the range between entities and clusters (the most used measurement is the standard Euclidean Distance) as in equation (1) [25]:

$$f = \sum_{i=1}^{K}\sum_{j=1}^{N}\left\|x_j - C_i\right\|^2 \quad j \in G_i \tag{1}$$

where K is the number of clusters, N is the number of objects, x_j is the coordinate of object j, C_i is the coordinate of the cluster i and G_i is the group of objects that belong to cluster *i*. The algorithm shifts the cluster in space to reduce the square distances within the cluster. The positions of all objects belonging to each cluster are recalculated by averaging. Calculation of the center uses equation (2):

$$C_i = \frac{1}{|G_i|}\sum_{j=1}^{N} X_j \quad j \in G_i \tag{2}$$

where G_i is the number of objects in the cluster *i*. The algorithm begins with a random set of the $C_{i\ cluster}$'s initial K center points(*i*=1,...,*K*), which are the present centroids.

Finding the optimal *k* number is necessary as the K-Mean requires a good initialization to prevent getting stuck at a wrong local optimal solution. The number of clusters is determined by running the algorithm several times and picking the appropriate clusters based on a few validity criteria or automatically identifying them using practical approaches or standards. Likewise, the cluster centers can be altered and adjusted numerous times by executing the procedure [26]. Currently, three approaches can be used to validate the clusters. These include external, internal, and relative cluster validity indexes [27]. Clustering validation is a strategy for identifying a set of clusters that best fits natural partitions without the need for class information. To date, no widely used index is acceptable for all data sets. As a result, determining a good validity index to discover an optimal *k* cluster number remains a challenging task [28].

There is no standard selection process that exists to choose the optimal value of the *k* number. Thus, a couple of methods, the Elbow and Silhouette methods, are chosen. These two generally help select the number of displayed clusters [29]. Both determine the validity of a clustering method based on the internal validity index [30]. For the external validity index, V-measure is considered for this research. In order to implement the methods, software programming Python 3.8, with the language programming of Python, is used.

D. Evaluate Clusters Performances

The V-measure score indicates how well the homogeneity and completeness requirements have been met. The harmonic mean of various homogeneity and completeness ratings is used to calculate it [31]. Homogeneity is satisfied when each cluster in the clustering result contains only data points members of a single class. Completeness is satisfied if all data points in a particular class are constituents of the same cluster in the clustering result; then, the clustering is complete [23].

V-measure is used because external validity indices are frequently used to compare cluster results to a previously established partitioning scheme [32]. Precisely for this study, the number of clusters suggested from Elbow and Silhouette methods. Completeness score is favored by values larger than 1, whereas homogeneity score is favored by values less than 1. Both ratings are weighted equally for this research implementation. The final score is between 0 and 1, with 1 denoting perfect clustering [27]. The method that produced the highest score is chosen as the best optimal *k* number.

E. Calculate Intracluster Distance

Intracluster distances are the sum of distances between data points and their corresponding parent cluster centroids, with smaller values indicating more grouping compactness [33]. A software called Rapidminer is used to help calculate the intracluster distance once the clusters have been produced. It shows a list of each clusters intracluster distances and, from there, can determine which best cluster to choose. Equation (3) shows the formula intracluster distance [34].

$$\frac{1}{n^*}\sqrt{\sum_{i=1}^{n^*}\left\|x_i - c\right\|^2} \tag{3}$$

where n^* represent the total number of data, i represent the cluster, x_i represents data, and c denotes the cluster center.

Fig. 3 shows the design to produce the intracluster distance values. The data results will be retrieved and read first in the process section. Then, 'Select attribute' selects a subset of attributes from the data results and removes the other attributes. Next, 'Clustering' is basically to perform the K-Mean algorithm. Lastly, 'Performance' will evaluate centroid-based clustering methods' performance and deliver a list of performance criteria values based on cluster centroids. Once the process is executed, the output will list out the intracluster distance for each cluster produced.

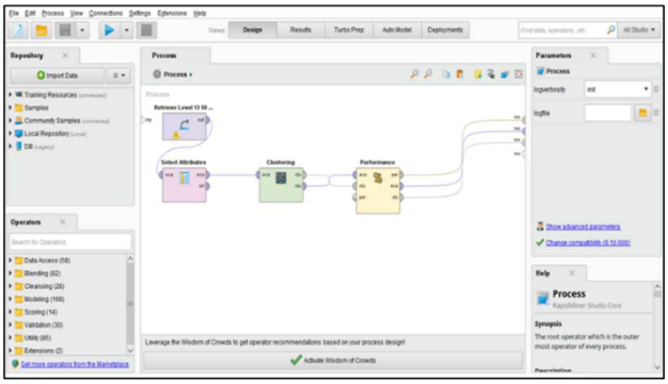


Fig. 3 The design to produce the intracluster distance values

F. Determine The Best Evacuation Model

The cluster with the lowest intracluster distance best determines the best evacuation model. The lowest time taken to exit the building is searched and chosen. The chosen time taken is determined by the best evacuation model representing the simulation software that produced the lowest time taken.

III. RESULTS AND DISCUSSIONS

This section discussed the developed indoor evacuation assessment algorithm, including the V-measure score, intracluster distance, and find the best evacuation model algorithms. Also, validate the testing and evaluation of the assessment algorithm where a validation form is used.

A. Indoor Evacuation Assessment Algorithm

Algorithm 1 shows the pseudocode of the indoor evacuation assessment algorithm. It starts with initializing the necessary variables. Then, input the data in attributes where the value is based on the researchers themselves. The output is the smallest time taken from the simulation results. The procedure includes choosing which simulation software to use to implement the data attributes and running the simulation. Once all the time taken results are obtained from the three-simulation software, we can implement the K-Mean coding. From numbers 30 to 35, the coding shows that the K-Mean algorithm is used first and looped 2 to 10 times. Each k number cluster with its corresponding clustering results is obtained and saved during this process. It is used during pseudocode numbers 33 and 34 to obtain the suggested optimal k number for Elbow and Silhouette method. The coding then displays the suggested optimal k number visualized in a graph, evaluating and choosing the optimal k number. After that, the algorithm continues by calling and running, in corresponding order, the algorithm for V-measure Score, Intracluster Distance, and Find Best Evacuation Model. The assessment algorithm then ends.

Algorithm 1: Indoor Evacuation Assessment Algorithm

```
1.  Start
2.  Initialize X = {x1, x2,…,xn};  //set of n simulation time
                                     taken results
3.  Initialize k = 0;              //number of clusters
4.  Initialize sse, slc = {},{};   //sse for elbow and slc for silhouette
5.  Initialize kmeans = 0;
6.  Initialize n_clusters = 0;
7.  Initialize max_iter = 0;
8.  Initialize random_state = 0;
9.  Initialize cluster = 0;
10. Input:
11.     agents;     //Number of agents and number of agents
                     must be above 0
12.     behaviours;     //Agents' behaviour and identify
                         "Grouped" or "Scattered"
13.     rooms; //Room size and total room size of the floor
                must be above 0
14.     doors;//Number of doors and total number of doors
               agents used must be above 0
15.     staircases; //Number of staircases and total number
                     of staircases agents used must be
                     above 0
16.     blockage;  //Blockage and identify either "Yes" or
                    "No"
17.     exit;   //Number of exits and total number of exits
                 available must be above 0
18. Output: Smallest time taken (s) results from simulation.
19. Procedure:
20.     switch (evacuation model)      //Select and run
                                        evacuation
                                        model simulation
                                        (Pathfinder,PedGo,
                                        AnyLogic)
21.     case "Pathfinder": Pathfinder (agents, behaviour,
         rooms, doors, staircases, blockage, exits);
                              //run Pathfinder simulation
22.     break;
23.     case "PedGo": PedGo (agents, behaviour, rooms,
           doors, staircases, blockage, exits);
                              //run PedGo simulation
24.     break;
25.     case "AnyLogic": AnyLogic(agents, behaviour,
           rooms, doors, staircases, blockage, exits);
                              //run AnyLogic simulation
26.     break;
27.     default Display("Invalid input!");
28.     break;
29. endcase;
30. for k in range (2,10)   //for K-Mean function and set
                             clusters
                         range between 2 to 10
31.     kmeans = KMeans (n_clusters = k, max_iter = 1000,
       random_state = 10).fit(X);
                          //run K-Mean and fit() used to
                           clear any attributes in the
                           estimator and perform
                           parameter and data
                           validation
32.     clusters = kmeans.labels;
33.     sse[k] = kmeans.inertia; //Optimal k number cluster
                                  chosen based on elbow
                                  point
34.     slc[k] = silhouette_score(X,clusters);
                          //Optimal k number cluster
                           chosen based on highest
                           silhouette score
35. endfor
36. Run V-measure score(optimal_k_elbow,optimal_k_silhouette);
                       //to evaluate optimal k number
                        cluster for elbow and silhouette
                        were the V-measure score is
                        compared, and the highest score
                        is chosen as optimal k number
37. Run intracluster distance(true_optimal_k, X);
    //calculate intracluster distances and true optimal k is
     obtained from V-measure score
38. Run find best evacuation model(lowest);
     //to find the lowest  time taken from each of the chosen
      clusters and the chosen cluster is based on the
      lowest intracluster distance
39. End
```

B. V-measure Score Algorithm

Algorithm 2 shows the pseudocode's V-measure score, which initializes the needed variables. The output is the V-measure score of optimal k number from the Elbow and

Silhouette method. The procedure starts by reading the excel file containing the simulation data results. Then, from numbers 18 to 27, the algorithm calculates the V-measure score. Lastly, numbers 28 to 30 compare the scores and return the highest V-measure score value. The optimal *k* number that represents the highest V-measure score is chosen to be used for K-Mean.

Algorithm 2: V-measure score

```
1.  Start
2.  Initialize df = pd.read_csv();       //read attributes and
3.                                        data in excel
4.  Initialize y = 0;
5.  Initialize X = 0;
6.  Initialize e_kmeans = 0;
7.  Initialize s_kmeans = 0;
8.  Initialize e_labels = 0;
9.  Initialize s_labels = 0;
10. Initialize e_clusters = optimal_k_elbow;
11. Initialize s_clusters = optimal_k_silhouette;
12. Initialize elbow_score = 0;
13. Initialize silhouette_score = 0;
14. Initialize true_optimal_k;
15. Output: V-measure score for Elbow and Silhouette
16. Procedure:
17. y = df['ID'];              //separate dependent variables
18. X = df.drop['ID', axis=1];
    //separate independent variables and .drop() to drop
    column
19. e_kmeans = KMeans(e_clusters);  //set cluster to
                                     certain number
                                     to determine the
                                     score for elbow
20. s_kmeans = KMeans(s_clusters);  //set cluster to
                                     certain number
                                     to determine the
                                     score silhouette
21. e_kmeans.fit(X);  //train cluster model for elbow
22. s_kmeans.fit(X); //train cluster model for silhouette
23. e_labels = e_kmeans.predict(X);   //.predict() use the
                                       learned parameters
                                       by .fit() in to predict
                                       on new and unseen
                                       test data points for
                                       elbow
24. s_labels = s_kmeans.predict(X);   //.predict() use the
                                       learned parameters
                                       by .fit() in to predict
                                       on new and unseen
                                       test data points for
                                       silhouette
25. v_measure_score(y,e_labels); //produce V-measure
                                  score for elbow
26. elbow_score = v_measure_score(y,e_labels);
27. v_measure_score(y,s_labels);   //produce V-measure
                                    score for silhouette
28. silhouette_score = v_measure_score(y,s_labels);
29. if elbow_score bigger than silhouette_score
    //compare V-measure score between elbow and
    silhouette
30.       return true_optimal_k = e_clusters;
31. else return true_optimal_k = s_clusters;
32. End
```

C. Intracluster Distance Algorithm

Algorithm 3 shows the intracluster distance pseudocode, which initializes necessary variables. The output is the lowest intracluster distance value. The procedure begins with calculating the intracluster distance of each cluster. Each value is stored in an array as in step number 18. The algorithm then compares each value and returns the lowest intracluster distance value.

Algorithm 3: Intracluster Distance

```
1.  Start
2.  Initialize n*;                           //total number of data
3.  Initialize i = 0;
4.  Initialize k = true_optimal_k;
5.  Initialize x_i = X          //set of n simulation time taken results
6.  Initialize c;                                   //cluster center
7.  Initialize sum = 0;
8.  Initialize b = 1/n*;
9.  Initialize double[] intraDistance = new double(k);
                     //intracluster distance are put in array
10. Initialize lowest = intraDistance[0];
11. Output: Lowest intracluster distance
12. Procedure:
13. for i in range (0, k)      //compare between the intracluster
                                 distance
14.       Sqrt for j in range (1, n*+1)
15.             sum = || x_i – c ||**2;
16.       endfor
17.       b * sum;
18.       intraDistance[i] = b * sum;
19.       if lowest smaller than intraDistance[i]
20.                lowest = intraDistance[i];
21.       endif
22. endfor
23. return lowest; //return lowest intracluster distance with its
                     associated cluster
24. End
```

D. Find Best Evacuation Model Algorithm

Algorithm 4 shows the find best evacuation model pseudocode, which begins with initializing the essential variables. The output is the lowest time taken. The algorithm then proceeds to loop each chosen cluster. While looping, it compares the cluster's time taken and inserts the lowest time taken in an array. Lastly, the algorithm displays each customer's lowest time taken, and the algorithm ends.

Algorithm 4: Find the Best Evacuation Model

```
1.  Start
2.  Initialize TC      //total number of the chosen cluster
3.  Initialize T[j];    //an array of time taken of the chosen cluster
4.  Initialize lowest_time = T[0];
5.  Initialize double[] LowestTimeTaken = new double(TC);
6.                          //the lowest time taken are put in array
7.  Output: Lowest Time Taken
8.  Procedure:
9.  for i = 0; i is smaller than equal to TC; i++
                                   //looping each chosen cluster
10.       for j = 1 is smaller than equal to T.length; j++
                               //compare time taken in cluster
                               and T.length is the size of time
                               taken in the  cluster
11.             if lowest_time smaller than T[j]
12.                  lowest_time = T[j];
13.             endif
14.       endfor
15. LowestTimeTaken[i] = lowest_time; //lowest_ time is put in
                                                 array
16. Endfor
17. display LowestTimeTaken[i];
18. End
```

Fig. 4 shows the detailed flowchart for the indoor evacuation assessment algorithm. The development of the indoor evacuation assessment algorithm is critical. To be precise, once the values of attributes have been inserted, they can be used in the chosen evacuation simulations. The three-simulation software is used. It helps calculate the time taken for the agents to evacuate the building. Once the results are produced, the K-Mean clustering algorithm is implemented. At this stage, several steps are taken to get accurate results. The steps include finding the optimal *k* number, validating *k* number again with the V-measure score, and finding the smallest value of the intracluster distance between clusters. Lastly, the assessment algorithm displays the best evacuation model.

F. Testing and Evaluation

Testing the assessment algorithm is conducted after development. In this phase, the algorithm is used to check K-Mean clustering results. Any anomalies found in the algorithm will be corrected and tested again unless the anomalies can be ignored. The assessment algorithm's evaluation is then conducted to validate how effective the algorithm is in getting the best evacuation model for this research. The expert will validate the validation to see if the assessment algorithm works. Those experts include algorithm experts from Universiti Teknologi MARA lecturers and Malaysia Fire and Rescue Station safety officers. Fig. 5 shows the validation questionnaire form used for the validation by the experts.

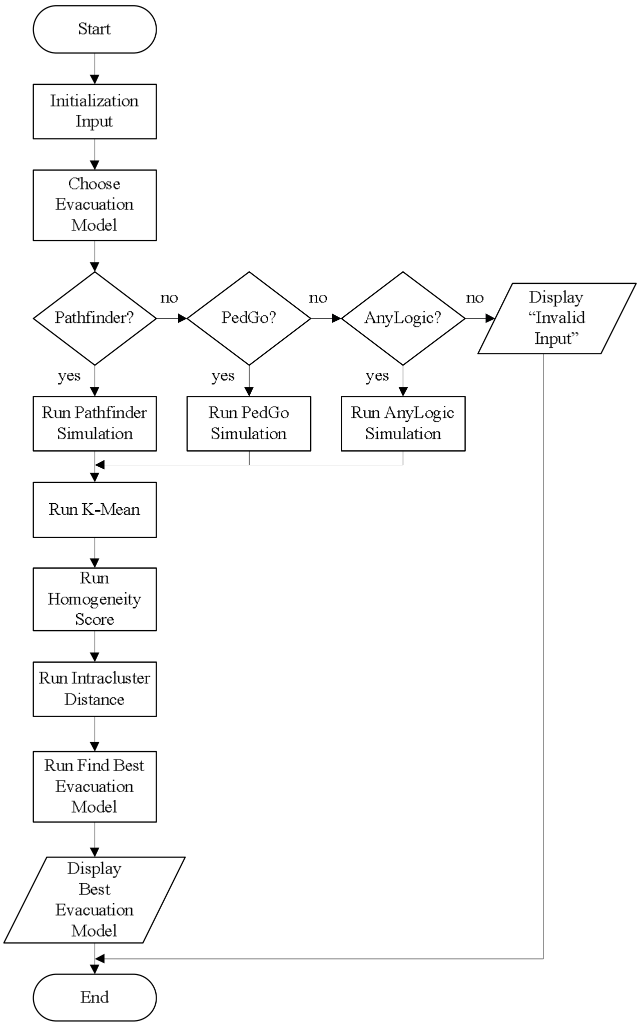


Fig. 4 Flowchart indoor evacuation assessment algorithm

Indoor Evacuation Model Assessment Algorithm Validation Form Questionnaire

Name : ______________________

IC : ______________________

Occupation : ______________________

Date : ______________________

Direction: Please read and validate the questionnaire by checking the choices which best describe your honest opinion towards the questionnaire. Your comments and suggestions for the improvement of the questionnaire will be highly appreciated.

Scale: 5- Excellent 4- Very Good 3- Good 2- Fair 1- Poor

No		5	4	3	2	1
1	The assessment algorithm is simple and easy to understand.					
2	The assessment algorithm is presented and organized in logical manner.					
3	The attributes used in the assessment algorithm are appropriate.					
4	The assessment algorithm reaches the objectives to assess the best evacuation model.					

Comments : ______________________

Signature and Official Stamp : ______________ **Date** : ______________

Fig. 5 Validation questionnaire form

IV. CONCLUSION

This study presented the phases involved in developing an indoor evacuation assessment algorithm. It is proposed to help the building owners choose the most suitable evacuation model before implementing it. The involved process in choosing the evacuation models and clustering algorithm to be used is the assessment algorithm's heavy influence. For the evacuation model, the commonly used type is the microscopic model, and ABM, SFM, and CA are the most popular used existing evacuation models. The assessment algorithm uses simulation software associated with the existing evacuation models: Pathfinder, AnyLogic, and PedGo. For the clustering algorithm, K-Mean was used to cluster the simulation results. K-Mean was found to cluster large datasets, has less time complexity, and has a less computational cost. For future work, it is recommended to use other clustering algorithms. K-Mean might be the most straightforward, but it has some disadvantages. For instance, it was mentioned earlier that K-Mean can get stuck to local optima. Therefore, other clustering algorithms can be adapted to cluster the results more accessible and faster.

ACKNOWLEDGMENT

The research is supported by the Ministry of Education Malaysia (MoE) and Universiti Teknologi MARA through the Fundamental Research Grant Scheme (FRGS-RACER) (600-IRMI/FRGS-RACER 5/3 (124/2019)).

REFERENCES

[1] G. M. Ventura, "Patient Evacuation Resource Classification System (PERCS) for residential healthcare facilities : Patient classification system translatable to healthcare evacuation protocols, system

modeling, and transportation resources," The George Washington University, 2017.
[2] H. Gao, B. Medjdoub, H. Luo, H. Zhong, B. Zhong, and D. Sheng, "Building evacuation time optimization using constraint-based design approach," *Sustainable Cities and Society*, vol. 52, no. 4, 2020.
[3] Y. Chen, C. Wang, H. Li, J. B. H. Yap, R. Tang, and B. Xu, "Cellular automaton model for social forces interaction in building evacuation for sustainable society," *Sustainable Cities and Society*, vol. 53, no. September 2019, 2020.
[4] Y. Jiang, B. Chen, X. Li, and Z. Ding, "Dynamic navigation field in the social force model for pedestrian evacuation," *Applied Mathematical Modelling*, vol. 80, pp. 815–826, 2020.
[5] P. Kontou, I. G. Georgoulas, G. A. Trunfio, and G. C. Sirakoulis, "Cellular automata modelling of the movement of people with disabilities during building evacuation," *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, pp. 550–557, 2018.
[6] N. A. A. Bakar, K. Adam, M. A. Majid, and M. Allegra, "A simulation model for crowd evacuation of fire emergency scenario," *ICIT 2017 - 8th International Conference on Information Technology, Proceedings*, pp. 361–368, 2017.
[7] F. Martinez-Gil, M. Lozano, I. García-Fernández, and F. Fernández, “Modeling, evaluation, and scale on artificial pedestrians: A literature review,” *ACM Computing Surveys*, vol. 50, no. 5, 2017.
[8] R. Ming and X. Peng, "Study on the social force model of personnel evacuation in large stadiums," *14th International Conference on Services Systems and Services Management, ICSSSM 2017 - Proceedings*, pp. 1–5, 2017.
[9] M. Shi, E. W. M. Lee, and Y. Ma, "A novel grid-based mesoscopic model for evacuation dynamics," *Physica A: Statistical Mechanics and its Applications*, vol. 497, pp. 198–210, 2018.
[10] I. Sakour and H. Hu, "Robot-assisted crowd evacuation under emergency situations: A survey," *Robotics*, vol. 6, no. 2, 2017.
[11] Y. Li, M. Chen, X. Zheng, Z. Dou, and Y. Cheng, "Relationship between behavior aggressiveness and pedestrian dynamics using behavior-based cellular automata model," *Applied Mathematics and Computation*, vol. 371, 2020.
[12] L. Fayez, "Modeling family behaviours in crowd simulation," Qatar University, 2017.
[13] K. Fisher-Vanden and J. Weyant, "The evolution of integrated assessment: Developing the next generation of use-inspired integrated assessment tools," *Annual Review of Resource Economics*, vol. 12, pp. 471–487, 2020.
[14] G. E. Metcalf and J. H. Stock, "Integrated assessment models and the social cost of carbon: A review and assessment of U.S. experience," *Review of Environmental Economics and Policy*, vol. 11, no. 1, pp. 80–99, 2017.
[15] A. H. A. Halim, K. A. F. A. Samah, Z. Ibrahim, and R. Hamzah, "Conceptual framework for intelligent indoor evacuation model assessment algorithm using integrated assessment model," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 1.4, pp. 289–294, 2020.
[16] S. Kasereka, N. Kasoro, K. Kyamakya, E. F. Doungmo Goufo, A. P. Chokki, and M. V. Yengo, "Agent-based modelling and simulation for evacuation of people from a building in case of fire," *Procedia Computer Science*, vol. 130, pp. 10–17, 2018.
[17] Y. Li, H. Liu, G. peng Liu, L. Li, P. Moore, and B. Hu, "A grouping method based on grid density and relationship for crowd evacuation simulation," *Physica A: Statistical Mechanics and its Applications*, vol. 473, no. 88, pp. 319–336, 2017.
[18] W. Liu and D. Parhizgar, "Evaluating classroom evacuation with crowd simulation," KTH Royal Institute of Technology, 2018.
[19] E. Ronchi and D. Nilsson, "Fire evacuation in high-rise buildings: a review of human behaviour and modelling research," *Fire Science Reviews*, vol. 2, no. 1, p. 7, 2013.
[20] A. Garcimartin, D. Maza, J. M. Pastor, D. R. Parisi, C. Martin-Gomez, and I. Zuriguel, "Redefining the role of obstacles in pedestrian evacuation," *J. Phys. Energy*, vol. 2, no. 1, pp. 0–31, 2020.
[21] K. Zhu, Y. Yang, and Q. Shi, "Study on evacuation of pedestrians from a room with multi-obstacles considering the effect of aisles," *Simulation Modelling Practice and Theory*, vol. 69, pp. 31–42, 2016.
[22] J. Chen, D. Liu, S. Namilae, S. A. Lee, J. E. Thropp, and Y. Seong, "Effects of exit doors and number of passengers on airport evacuation effeciency using agent based simulation," *International Journal of Aviation, Aeronautics, and Aerospace*, vol. 6, no. 5, 2019.
[23] A. Vysala and D. J. Gomes, "Evaluating and validating cluster results," *Computer Science & Information Technology (CS & IT)*, pp. 37–47, 2020.
[24] Nurhayati, N. S. Sinatrya, L. K. Wardhani, and Busman, "Analysis of k-means and k-medoids's performance using big data technology," *2018 6th International Conference on Cyber and IT Service Management, CITSM 2018*, pp. 1–5, 2019.
[25] A. B. S. Serapião, G. S. Corrêa, F. B. Gonçalves, and V. O. Carvalho, "Combining k-means and k-harmonic with fish school search algorithm for data clustering task on graphics processing units," *Applied Soft Computing Journal*, vol. 41, pp. 290–304, 2016.
[26] A. Pugazhenthi and L. S. Kumar, "Selection of optimal number of clusters and centroids for k-means and fuzzy c-means clustering: A review," *Proceedings of the 2020 International Conference on Computing, Communication and Security, ICCCS 2020*, pp. 5–8, 2020.
[27] V. B. B. Anguiano, "Integration and visualization of sparse-grid based clustering methods in the SG++ datamining pipeline," Technical University of Munich, 2019.
[28] X. Li, W. Liang, X. Zhang, S. Qing, and P. C. Chang, "A cluster validity evaluation method for dynamically determining the near-optimal number of clusters," *Soft Computing*, vol. 24, no. 12, pp. 9227–9241, 2020.
[29] J. Guo, "Developing a visualization tool for unsupervised machine learning techniques on *Omics data," 2018.
[30] D. M. Saputra, D. Saputra, and L. D. Oswari, "Effect of distance metrics in determining k-value in kmeans clustering using elbow and silhouette method," vol. 172, no. Siconian 2019, pp. 341–346, 2020.
[31] P. Mengoni, A. Milani, and Y. Li, *Clustering students interactions in eLearning systems for group elicitation*, Springer International Publishing, 2018.
[32] C. Tomasini, E. N. Borges, K. Machado, and L. Emmendorfer, "A study on the relationship between internal and external validity indices applied to partitioning and density-based clustering algorithms," *2017 19th International Conference on Enterprise Information Systems (ICEIS)*, vol. 1, pp. 89–98, 2017.
[33] S. Sengupta, S. Basak, and R. A. Peters, "Data clustering using a hybrid of fuzzy c-means and quantum-behaved particle swarm optimization," *2018 IEEE 8th Annual Computing and Communication Workshop and Conference, CCWC 2018*, vol. 2018-Janua, pp. 137–142, 2018.
[34] S. Panda, S. Sahu, P. Jena, and S. Chattopadhyay, "Comparing fuzzy-c means and k-means clustering techniques: A comprehensive study," *Advances in Intelligent and Soft Computing*, vol. 166 AISC, no. VOL. 1, pp. 451–460, 2012.