IAES International Journal of Artificial Intelligence (IJ-AI)

IAES International Journal of Artificial Intelligence (IJ-AI), ISSN/e-ISSN 2089-4872/2252-8938 publishes articles in the field of artificial intelligence (AI). The scope covers all artificial intelligence (AI) and machine learning (ML) areas and theirs applications in the following topics: neural networks; fuzzy logic; simulated biological evolution algorithms (like genetic algorithm, ant colony optimization, etc); reasoning and evolution; intelligence applications; computer vision and speech understanding; multimedia and cognitive informatics, data mining and machine learning tools, heuristic and AI planning strategies and tools, computational theories of learning; technology and computing (like particle swarm optimization); intelligent system architectures; knowledge representation; bioinformatics; natural language processing; multiagent systems; supervised learning; unsupervised learning; deep learning; big data and AI approaches; reinforcement learning; and learning with generative adversarial networks; etc. This journal is indexed in Scopus and all published papers since 2018 issues were included in scopus.com.

Focus and Scope

The IAES International Journal of Artificial Intelligence (IJ-AI), ISSN/e-ISSN 2089-4872/2252-8938 covers all topics of artificial intelligence and soft computing and their applications, including but not limited to:

- Neural networks
- Reasoning and evolution
- Intelligent search
- Intelligent planning
- Intelligence applications
- Computer vision and speech understanding
- Multimedia and cognitive informatics
- Data mining and machine learning tools, heuristic and AI planning strategies and tools, computational theories of learning
- Technology and computing (like particle swarm optimization); intelligent system architectures
- Knowledge representation
- Bioinformatics
- Natural language processing
- Automated reasoning
- Logic programming
- Machine learning
- Visual/linguistic perception
- Evolutionary and swarm algorithms
- Derivative-free optimisation algorithms
- Fuzzy sets and logic
- Rough sets
- Simulated biological evolution algorithms (like genetic algorithm, ant colony optimization, etc)
- Multi-agent systems
- Data and web mining
- Emotional intelligence
- Hybridisation of intelligent models/algorithms
- Parallel and distributed realisation ofintelligent algorithms/systems
- Application in pattern recognition, image understanding, control, robotics and bioinformatics
- Application in system design, system identification, prediction, scheduling and game playing
- Application in VLSI algorithms and mobile communication/computing systems

IAES International Journal of Artificial Intelligence (IJ-AI)

IAES International Journal of Artificial Intelligence (IJ-AI)
Vol 11, No 1: March 2022

Table of Contents

IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 11, No. 1, March 2022, pp. 1~12
ISSN: 2252-8938, DOI: 10.11591/ijai.v11.i1.pp1-12

Inspection based risk management of electric distribution overhead lines

Adnan Hasan Tawafan[1], Dhafer Mayoof Alshadood[2], Fatima Kadhem Abd[1]
[1]Al-Furat Al-Awsat Technical University, Technical Institute Karbala, Karbala, Iraq
[2]Southern Technical University, Nasiriyah Technical Institute, An-Nashiriyah, Iraq

Article Info

Article history:

Received Jul 27, 2021
Revised Dec 24, 2021
Accepted Jan 4, 2022

Keywords:

Distribution network
Reliability indices
Risk management
Shortest path problem

ABSTRACT

This paper introduces a comprehensive process based on inspection for determination of medium voltage (MV) overhead line condition and it was tried that all factors influencing the outage of distribution network integrated into one index that called condition index. A condition based failure rate model has been proposed and its unknown parameters are calculated based on historical data. Shortest path problem (SPP) model has been proposed for the long term scheduling of maintenance and reinvestment. Objective function includes sum of the reinvestment, maintenance costs, failure costs and energy not supplied (ENS) costs with considering budget and labor constraints. Finally, as a result of this research, optimal combination of various actions such as reinvestment, preventive maintenance (PM) and tree trimming and it's scheduling has been determined over the ten and five-year horizon. Results confirmed acceptable performance of proposed method because of compliance with actual condition and engineering judgment.

Corresponding Author:

Adnan Hasan Tawafan
Al-Furat Al-Awsat Technical University, Technical Institute Karbala
Iraq
Email: towfana@yahoo.com

1. INTRODUCTION

Overhead lines are important components in the distribution of electrical energy. According to statistics, a large percentage of electricity costs, and most costumers' outage, occur in the distribution network. In fact, about 80% of the customers' outage are due to a failure in the distribution network [1]. Distribution networks in urban and rural areas often consist of overhead feeders. Poles, cross arms, insulators, jumpers, conductors and transformers are overhead distribution network equipment, thus forming a complex and repairable system. Conditions of all these component should be monitored, because the condition of each component will influence the overall condition of an overhead line [2]. Distribution companies strive to reduce their operation costs for achieving greater profits and maintaining their competitiveness [3]. In this regard, it is of great importance to reduce maintenance and repair costs, which constitute a significant part of the costs of operating the distribution network. Outages can be divided into four major categories. First, the failures that result from aging and failure of equipment and are related to the health of equipment and their service life. The second category is caused by wrong design and external factors and not related to the life and health of the equipment. The third category is due to the impact of trees and because the conductors are bare and also lack of timely trimming of the trees. Finally, the fourth category is caused by other factors such as human error and unknown factors. This information indicates that maintenance can only prevent one-third of the outages. According to experts, more than half of the unknown outage are most likely due to external factors. Therefore, to prevent the occurrence of nearly half of outages, the design of the network must be changed and retrofitted against external factors. Therefore, failure rate modeling based on the age of the equipment does not seem to be realistic and is therefore a perfect model that takes into account the effects of environmental and external factors. As well as reflect the effect of performing corrective/preventive maintenance and reinvestment actions in failure rate improvement.

Various methods have been used for evaluating and optimizing the risk, reliability and maintenance scheduling in distribution networks. As demonstrated in [4] utilized dynamic scheduling for long term maintenance management [5], [6], decision tree [7], [8], cost-benefit ratio [9]–[11], markov method [12], monte carlo [13]–[15]. Are the methods that risk and reliability of a network have been evaluated by them. The shortest path problem (SPP) method has been used in this study. SSP has been used for solving many different problems in the power system [16], [17]. Five actions are proposedfor managing risk in distribution networks, one option is to make no action during the network scheduling period, the next action is reinvestment for changing the design of the network and three actions are proposed for reinvestment.

2. RISK MANAGEMENT ACTIONS

The purpose of this study is to investigate the implementation of risk management in real distribution network. Risk management means reducing or keeping the risk at the optimal level using available and possible tools. But what tools do we have for risk management? Maintenance is only one of the risk management action, other tools such as replacement and reinvestment are available, and the choice of each of these action in each interval depending on the goals and constraints of each action. In describing the objectives, the expected level of risk can be mentioned. This issue that how much we want to reduce the risk depends on the goals and the prospects. Among the limitations are budget constraints and human resources that are influencive in achieving goals because we may not achieve the desired level of risk given these constraints.

Our available and proposed actions for managing risk in distribution networks are illustrated in Figure 1. These actions include five categories. One option is to make no action during the network scheduling period. As a result, no budget are incurred, but the network situation worsens and the failure rate increases, imposing additional costs on the distribution companies, including the cost of the failures and energy not supplied. Another action is network maintenance. This means that we reduce the risk of network outages without drastically changing the network design by taking preventive and corrective actions. Maintenance can include tree trimming, minor maintenance and major maintenance. Therefore, the cost and effectiveness of each level of maintenance vary. The next action is reinvestment for changing the design of the network. Sometimes no level of maintenance can fix some of the outage, so it is essential that the design of the network be changed. For this purpose, based on the experiences of different countries and references, three actions are proposed for reinvestment as shown in the Figure 1.

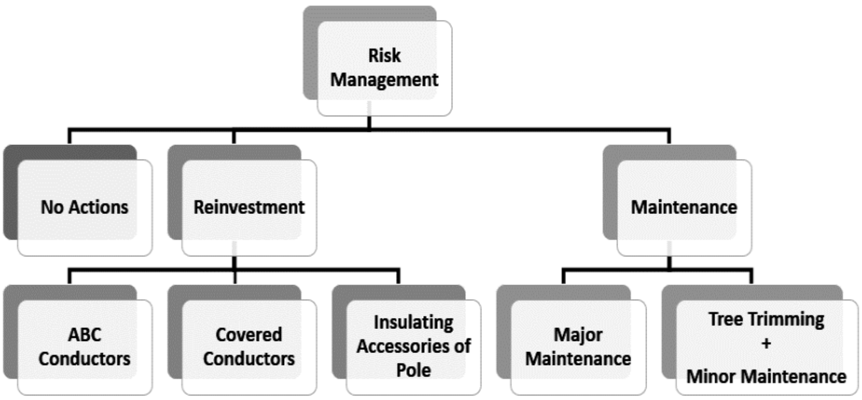


Figure 1. Risk management actions

3. RISK MANAGEMENT MODEL

The proposed risk management model is that we are faced with a choice of several actions in each period (e.g. a year). These actions include several maintenance actions and several investment actions that are selected according to the constraints and goals of the asset managers. The SPP is to find the optimum path from node 1 to end node n, which can be formulated as shown in the form of integer programming [18]:

$$min\, Z = \sum_{i=1}^{n}\sum_{i=1}^{n} c_{i,j}x_{i,j} \tag{1}$$

$$s.t \sum_{j=1}^{n} x_{i,j} - \sum_{k=1}^{n} x_{k,i} = \begin{cases} 1 & (i = 1) \\ 0\ (i = 2, 3, \dots, n-1) & \\ -1 & (i = n) \end{cases} \tag{2}$$

$$x_{i,j} = 0\ or\ 1 \quad \forall\, i, j \tag{3}$$

where Z is zone number. In the (2) and (3), we have: indices i, j, k: index of node (1, 2,…, n); parameters n: number of nodes and $c_{i,j}$: transmission cost of arc (i, j); decision variables (binary) $x_{i,j}$: the link on an arc (i, j).

3.1. Risk formulation

The factors that are defined under the risk index include the risk of human injury and breaking safety and/or environment laws and the risk arising from a customer power outage [19]. There are also other common categories in the electricity distribution system such as financial risk, safety risk, environmental risk, supply quality risk, reputation risk, vulnerability risk and law risk [20]. Some references also categorize risk factors as limited to financial, safety, network performance and environmental risk [21].

In analyzing and evaluating different risk reduction actions it should be considered, that performing on not performing of each action what effect does it have on the risk factors. In this paper, only financial risks are considered. One of the economic indices of analysis for long-term scheduling is net present value (NPV) that considering discount rate, transfers all annual costs to the present year. Since the objective function is financial, therefore the analysis can be made through net present value function as shown in (4):

$$C_{NPV} = \sum_{t=1}^{n} C \times (1 + DR)^{-n} \tag{4}$$

where DR is discount rate, the economic or financial risk itself includes reinvestment, renovation costs, operation, maintenance, repairs, damage costs, costumer outage cost such as energy not supplied (ENS) costs and revenue loss cost. Note, because many scheduled maintenance programs perform in a hot line, we neglect the costs of the planned outage.

$$Minimize\ CF = \sum_{z=1}^{NZ}\sum_{t=1}^{T}\sum_{k=1}^{ND} \Big((INV_z^k(t) \times \gamma_z^k(t)) + \big(C_z^{k,MT}(t) \times X_z^k(t)\big) + \big(C_z^F(t) \times \lambda_z(t)\big)\Big) \times (1 + DR^{-1}) \tag{5}$$

Where $\gamma_z^k(t)$ is binary variable related to investment actions, $X_z^k(t)$ is binary variable related to maintenance actions, $C_z^{k,MT}(t)$ is total cost related to the kth maintenance action of zone z in interval t and $C_z^F(t)$ is total cost related to the outage occurred on zone z in interval t.

3.1.1. Reinvestment cost

Three actions are proposed for reinvestment ABC conductor, cover conductor and covering accessories of pole. To calculate reinvestment costs, we need to know the amount of capital needed to run each action per kilometer. Therefore, the equation of investing in the distribution network are as shown in (6).

$$INV_z^k = inv^k \times L_z \tag{6}$$

where k=1: covering accessories of pole; 2: cover conductor; 3: ABC conductor.

3.1.2. Energy not served (ENS) cost

ENS means the amount of energy demand measured in MWh which is not supplied in a given zone and in a given time period due to insufficient resources to meet demand. ENS costs include budget and labor constraints. It can calculate as shown in (7) and (8).

$$ENS_z^{UP} = P_z \times D_z^{UP} \quad (7)$$

$$C_{ENS,z}^{UP} = C_{kWh} \times ENS_z^{UP} \quad (8)$$

3.1.3. Revenue loss cost

The economic or financial risk itself includes revenue loss cost. Total revenue loss cost is calculated by multiplying the ENS cost per unit by the quantity sold. It can calculate as shown in (9).

$$C_{Rev,z}^{UP} = C_E \times ENS_z^{UP} \quad (9)$$

3.1.4. Maintenance cost

The term maintenance cost refers to any cost incurred by an individual or business to keep their assets in good working condition. These costs may be spent for the general maintenance of items like price of one working hour necessary for the kth maintenance action or they may be used for repairs. It is calculate as shown in (10):

$$C_z^{k,MT}(t) = \left(\left(\left(C_{Labor}^{k,MT} \times Labor^{k,MT}\right) + C^{k,Mcomp}\right) \times L_z\right) \quad (10)$$

where k is 1 for minor maintenance, 2 for major maintenance or 3 for tree trimming.

$$C^{k,Mcomp} = \sum_{j=1}^{NM_k} C_j^{k,comp} \quad (11)$$

As demonstrated in (10) means that each maintenance action includes more minor items. As an example for minor maintenance, the cost of non-standard span correction, replacement of fittings and correction of defective tie conductor is calculated separately or for major maintenance, replacement cost of failed insulators and poles, and cost of failed equipment are calculated separately.

3.1.5. Failure costs

Failure costs include price of one working hour necessary for repairing fault occurred, number of working hours necessary for repairing the fault, unplanned ENS Cost per megawatt-hour for zone z and revenue lost due to unplanned ENS resulted from outage occurred on zone z. It is calculate as shown in (12).

$$C_z^F(t) = \left((C_{Labor}^R \times Labor^R) + C^{Rcomp}\right) + \left(C_{ENS,z}^{UP} + C_{Rev,z}^{UP}\right) \times (1+q)^{t-1} \quad (12)$$

3.1.6. Constraints

Constraints refer to total cost related to the kth maintenance action of zone z in interval t, total cost related to the outage occurred on zone z in interval t, capital Investment associated with the kth investment action of zone z in interval t, number of working hours necessary for the kth maintenance action, number of working hours available in interval t for tree trimming and number of working hours available in interval t for minor and major maintenance or repair. All constraints can calcutale as shown in (13)-(17).

$$\sum_{z=1}^{NZ}\sum_{k=1}^{ND}\left(\left(C_z^{k,MT}(t) \times X_z^k(t)\right) + \left(C_z^F(t) \times \lambda_z(t)\right)\right) \le Budget(t) \quad (13)$$

$$\sum_{z=1}^{NZ}\sum_{t=1}^{T}\sum_{k=1}^{ND}(INV_z^k(t) \times \gamma_z^k(t)) \times (1 + DR^{-1}) \le INV \quad (14)$$

$$\sum_{t=1}^{T}\sum_{k=1}^{ND}(\gamma_z^k(t)) \le 1 \quad (15)$$

$$\sum_{z=1}^{NZ}\sum_{k=1}^{MTi}\left(\left(Labor_z^{t,MT} \times X_z^k(t)\right) + \left(Labor_z^{t,MT} \times \lambda_z(t)\right)\right) \le Labor_{Available}^{MT} \quad (16)$$

$$\sum_{z=1}^{NZ}\sum_{k=1}^{ND}\left(Labor_k^{tt} \times X_z^{tt}(t)\right) \leq Labor_{Available}^{tt} \tag{17}$$

As demonstrated in (13) states that the total cost per year should not exceed the budget for that year. Equation (14) also emphasizes the limitation that the net present value of investment costs for all zones should be less than the amount of initial capital allocated to these zones. Also, in (15) notes that in long-term horizon (T), no more than one type of investment should be made for each zone. Equation (16) states that the number of man-hours for major and minor maintenance and emergency repairs per year should not exceed $Labor_{Available}^{MT}$, the number available per year. Likewise, in (17) the rate of man-hours for tree trimming per year should not exceed the number $Labor_{Available}^{tt}$.

4. STATISTICAL ANALYSIS OF OUTAGE DATA

Statistical data in distribution networks provides operators, planners, and asset managers with valuable information on the performance and condition of the network and equipment. This data is collected in a variety of ways from a combination of software data such as computerized maintenance management system (CMMS), geographic information system (GIS), and communication networks such as supervisory control and data acquisition (SCADA) [22]. This information helps us to find equipment reliability indices, worst failure type, frequent failures, geolocation of failures and many more.

5. CONDITION INDEX QUANTIFICATION

Technically the best way to do this is to continuous monitoring of equipment condition. But in addition to high costs, this method requires a lot of infrastructure and effort [22]. In many cases the health index is defined to determine the condition of an equipment. The health index includes all the factors that are effective in determining equipment condition [21].

As illustrated in Table 1 the criteria are decoupled into four categories of vulnerability index, tree condition index, minor health index, and major health index. This decoupling has been made due to the static nature and dynamics of some indices over time, as well as to better reflecting the impact of each of the risk management actions on overhead line condition.

Table 1. Check list for inspection of one zone of overhead lines

Criteria		Item	Item Weight	Item Score	Criteria Weight
Vulnerability Index	1	Vulnerability of poles to cars accident	w_{11}	s_{11}	W_1
	2	Vulnerability of conductors and pole accessories to birds	w_{12}	s_{12}	
	3	Vulnerability of conductors and pole accessories to other external things	w_{13}	s_{13}	
	4	Vulnerability of zone to lightning,	w_{14}	s_{14}	
		Weighted Mean of Scores	$X_1 = \frac{\sum_{j=1}^{M_1} w_{1j} \times s_{1j}}{\sum_{j=1}^{M_1} w_{1j}}$		$X_V = W_1 \times X_1$
Tree Condition Index	1	Vulnerability of conductors and pole accessories to trees	w_{21}	s_{21}	W_2
		Weighted Mean of Scores	$X_2 = \frac{\sum_{j=1}^{M_2} w_{2j} \times s_{2j}}{\sum_{j=1}^{M_2} w_{2j}}$		$X_{tree} = W_2 \times X_2$
Minor Health Index	1	Condition of tie wires on insulators	w_{31}	s_{31}	W_3
	2	Condition of connectors and jumpers	w_{32}	s_{32}	
	3	Condition of conductors distance	w_{33}	s_{33}	
	4	Condition of span sag	w_{34}	s_{34}	
		Weighted mean of scores	$X_3 = \frac{\sum_{j=1}^{M_3} w_{3j} \times s_{3j}}{\sum_{j=1}^{M_3} w_{3j}}$		$X_{HI\_minor} = W_3 \times X_3$
Major Health Index	1	Condition of poles (aging, corrosion)	w_{41}	s_{41}	W_4
	2	Condition of insulators	w_{42}	s_{42}	
	3	Condition of cut-outs, arresters and cable terminations	w_{43}	s_{43}	
	4	Condition of cross-arms	w_{44}	s_{44}	
		Weighted mean of scores	$X_4 = \frac{\sum_{j=1}^{M_4} w_{4j} \times s_{4j}}{\sum_{j=1}^{M_4} w_{4j}}$		$X_{HI\_major} = W_4 \times X_4$
Overall Condition Index			$X_t = \frac{X_V + X_{tree} + X_{HI\_minor} + X_{HI\_major}}{\sum_{j=1}^{4} W_j}$		

Another point in determining the factors affecting the overhead lines condition is the vulnerability of the overhead lines to external factors that are among the factors affecting the rate of outage or failure of the lines. The considering of this index was adopted after observing the extensive outages caused by these external factors in the statistical analysis. Table 1 was designed to determine the condition of a zone of feeders. The basis of zone division is the existence of maneuver points and the distance between two switching device. To quantification of the condition through this table, first each index is given a weight. A score will be assigned by the inspector for each index of each feeder zone. A score of 0 indicates the best and a score of 1 indicates the worst. The overall condition of each zone is calculated by the weighted average:

$$X_t = \frac{X_V + X_{tree}(t) + X_{HI\_major}(t) + X_{HI\_minor}(t)}{\sum_{j=1}^{4} W_j} \tag{18}$$

in the (18), X_t is the overall condition of each zone as a function of time, X_V is the condition index of vulnerability factor, $X_{HI\_minor}(t)$ is the minor health index and $X_{HI\_major}(t)$ is the major health index of overhead line's component. This will be more explained in next subsection.

5.1. Health index as function of time

In many cases the health index is defined to determine the condition of an equipment. The health index includes all the factors that are effective in determining equipment condition. Equipment such as poles and cross arms, fittings and jumpers, tie conductors, cutouts, and switches are subject to deterioration and aging. The following equation determines the health index of overhead line as a function of time [23], [24].

$$X_{HI}(t) = X_{HI_{t0}}\left(e^{\beta_{HI}(t-t_0)}\right) \tag{19}$$

In this equation, $X_{HI}(t)$ is an index of minor or major health at a particular time t in the future. $X_{HI_{t0}}$ is health index at time $t_0$and β_{HI} is a constant coefficient for scaling the equation.

5.2. Trees condition index as function of time

Trees are fast growing and can reach a critical point in about one to two years. If the tree branch approaches the bare lines, it is very likely that a failure will occur. In fact, the vulnerability of bare lines to trees is very high. Here, we want to estimate the vulnerability of lines to trees in each zone using a time-based model. If the condition index is between 0 (best) and 1 (worst) we assume that the condition will fall from 0.05 to 1 within 2 years. With this information, the unknown parameters of the (20) are obtained.

$$X_{tree}(t) = X_{tree_{t0}}\left(e^{\beta_{tree}(t-t_0)}\right) \tag{20}$$

In the (20), $X_{tree}(t)$ is the condition index of the trees in each zone at time $X_{tree_{t0}}$ is the current condition of the trees at time $t_0$and β_{tree} is the unknown variable of equation.

5.3. Scoring method of items

After determining the weight of each item, there is a need to set a framework for scoring the items. Since Table 1 is designed for the inspector, it should be as easy as possible for the inspector to assign a score to minimize errors and preferences in scoring. Therefore, for each item in Table 1, we will provide a selective framework for scoring. For example, the method of scoring an item is described next subsection.

5.3.1. Span condition

To determine the item's score, the inspector must specify the number of spans with a non-standard sag and length. There are an average of 200 poles in each zone. So, based on knowledge of utility expert, we give the worst score (number 1) to a zone that has more than 10 non-standard spans. The scoring framework for this index is given in Table 2.

5.4. Failure rate model

The failure rate is a key parameter in risk analysis of distribution network and is known as an index and measure of reliability of the equipment and is also very closely related to the condition of the equipment. The worst condition of an equipment will result in the highest failure rate in the equipment [25]. In general, it can be said the best failure rate model is a model that, besides accurately estimating the failure rate, can cover all the factors involved in the outage of feeder, as well as reflecting the impact of maintenance, replacement, and all overhead lines preventive actions. In [25], a practical model is presented that, unlike common modeling which are as a function of time and age of the equipment, it models the failure rate of each equipment based on its equipment condition. The model presented is an exponential function of the failure rate as a function of condition. Equation (21) illustrates this function [25].

$$\lambda(x) = \mathrm{A}e^{Bx} + C \qquad failure/year \tag{21}$$

Table 2. Span inspection check list for each zone

Score (0 to 1)	Non-standard span (span/zone)	Item
Span Condition	$x = 1$	0.05
	$2 \leq x \leq 3$	0.2
	$3 < x \leq 4$	0.4
	$4 < x \leq 6$	0.7
	$6 < x \leq 8$	0.8
	$8 < x \leq 10$	0.9
	$x > 10$	1

6. CASE STUDY

The studied network consists of 10 feeders (19 zone) in Mashhad, that there are different types of feeders among them in terms of load type and geographical area. The set of customers who have an outage after each failure of the feeder and separation of the faulty part is defined as a zone. Therefore, a zone is considered between the two automation switch.

6.1. Outage data analysis of case study

Mashhad electricity distribution company (MCDC) with yearly distribution of 6661094 MWh, is one of the leading and important distribution companies in the electricity industry of Iran. However, there are various failures on the network each year due to different reasons. According to the information recorded in the information and outage management system of MCDC, about 4604 failures has occurred in the medium voltage feeders in Mashhad in the last five years, out of which 4190 were in the overhead lines. The all of these failures in the overhead medium voltage feeders Leading to 3816 MWh energy not supplied.

6.2. Parameter calculation of failure rate model

To calculate the parameters A, B, and C of (21) for each zone, we will follow the description of [25]. First, we select the initial values from the [25] based on the length of each zone. But since the outage rate of each zone depends on the length of the feeder and various factors, it is therefore necessary to calibrate these values by using the Chi-square error method [25].

6.3. Present condition calculation and condition model estimation

To calculate the current condition of each equipment using Table 1, we obtain the weighted average of the characteristics and conditions of each zone in terms of its criteria and items. This average weight, which is the output of the table, indicates the condition of each zone under current conditions. The weight of the criteria and items is the same for its all feeders and related zones. To obtain minor and major β_{HI} from the (19), (22) and (23) is deduced respectively. Given the 20 year life span for the overhead lines and their initial state of 0.05, we obtain the β_{HI}. For $t_{0,major}$ and $t_{0,minor}$ it is reasonable to assume that they are equal to the year in which repairs are done.

$$\beta_{HI\_minor} = \frac{\ln\left(\frac{X_{HI_{t-minor}}}{X_{HI_{t0-minor}}}\right)}{t - t_{0,minor}} \tag{22}$$

$$\beta_{HI\_minor} = \frac{\ln\left(\frac{X_{HI_{t-minor}}}{X_{HI_{t0-minor}}}\right)}{t - t_{0,minor}} \tag{23}$$

Also to calculate β_{tree} by using the (20), (24) is concluded. We assume the worst-case condition $X_{tree} = 1$ for the bare conductor as 2 years, for the covered conductor as 4 years and for the ABC cables as 5 years and the initial condition value for all feeders is 0.05.

$$\beta_{tree} = \frac{\ln\left(\frac{X_{tree}}{X_{tree_{t0}}}\right)}{(t - t_0)} \quad (24)$$

6.4. Effect analysis of risk reduction actions

By performing each of the actions described above and applying the improvement effects of that actions on the condition index of the overhead lines in Table 2, the effectiveness of each actions is measured. There are now 5 actions outlined above, each applied separately and the rate of improvement in each condition index after applying actions is reported in Table 3.

Table 3. Rate of reduction of decoupled condition indices after implementation of actions

Risk Reduction Strategy	Rate of Reduction of Decoupled Condition Indices After Implementation of Actions (%)			
	Minor Health Index	Major Health Index	Trees Condition Index	Vulnerability Index
#1 (Action 1+2)	95	0	95	0
#2 (Action 3)	0	95	0	0
#3 (Action 1+2+4)	95	0	95	50
#4 (Action 3+5)	95	95	95	70
#5 (Action 3+6)	95	95	95	95

Maintenance actions

- Action 1: trees trimming to prevent encountering branches with trees.
- Action 2: minor repairs including replacement of damaged joints, correction of conductor spacing, correction of length and flash of span and tie conductors.
- Action 3: general repairs including equipment repair, cross-arm replacement, defective insulators replacement or repairing damaged poles, and replacement aged equipment.
- Reinvestment actions
- Action 4: implementing covering of pole’s accessories: this action directly improves the network's vulnerability to birds, external objects and trees.
- Action 5: replacing existing bare conductors with covered conductors: this action also has a direct impact on the amount of vulnerability to birds, external objects and trees. Of course, it is by itself vulnerable to storms and lightning. In the meantime and during implementation of the program, the connections, tie conductors, pole and cross-arms, the distance between the conductors and the bad span state will automatically corrected.
- Action 6: replacement of current bare conductors with ABC cable: ABC cable not only eliminates conditions such as poor span condition, conductor distance, tie conductors, jumper and cross-arm, it is durable against external factors, especially trees, storms and lightning is and does not have the disadvantages of a covered conductor.

Given that the overall condition of the feeder is resulted from the weighted average of the separated criteria, we therefore define that each action will only affect the self-relevant criteria. In accordance with Table 3, we define the strategies derived from the combination of the above actions. Rate of reduction of decoupled condition indices after implementation of actions are shown in Table 3 based on test and error using condition quantification framework in Table 1.

6.5. Implementation of proposed algorithm on the case study

In this section, the propsed model has been coded in the MATLAB [26] and solved by using priority-based genetic algorithm (priGA) 18] on the 10 feeders (19 zones) of the mashhad power distribution network described in the previous section, in 5-year horizons. The shortest path problem optimization model for 5-year horizon is shown in Figure 2. As it can be seen, we have 6 choices each year, depending on the specification of each zone and long-term scheduling, one of them will chose by the proposed algorithm. The chromosome dimensions of priGA after decoding for all zones will be $NZ \times T$, i.e. 19*5 for the 5-year horizon.

6.6. Problem assumption

Typically, in any research, there are assumptions to bring the situation closer to the real operating conditions. In the following, we will explain the assumptions we have made to solve this problem.

- Major maintenance will be carried out with the implementation of the ABC cable and coated conductor.
- As a result of implementation of ABC and covered conductor action, minor health index and tree condition index are also improved. In fact, we do not charge a separate cost to improve these indices.

– Vulnerability to trees becomes less after running ABC cable and coated conductor. Therefore, the index of tree condition later returns to their bad condition.
– Along with the implementation of the investment action for covering of pole's accessories, the tree trimming action and the minor maintenance action is done too.
– If we use ABC cable, there is no need for any major repairs until the end of the period.
– We are only allowed to use one investment action for each zone in each period.
– Partial repairs are also done with the trimming.

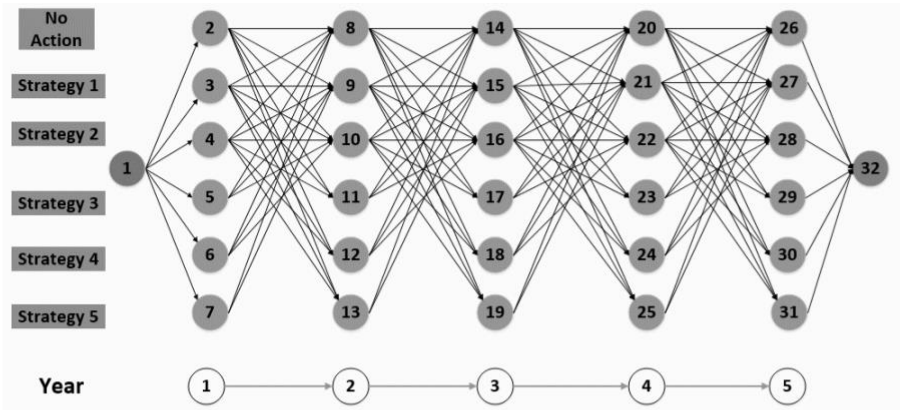


Figure 2. Shortest path problem for 5-year horizon

6.7. Simulation result in 5 year horizon

Simulation of the shortest path problem for finding the optimal strategy over the 5 years, is done based on model shown in Figure 2. In this problem, we have 6 options to choose each year. One option is to do no action on each zone, and the other 5 options include 5 different actions to improve the condition of each zone. By choosing an option for a year, for the following year only options can be selected that have a branch with the previous year option. In the following, we will examine the problems with considering constraints such as budget and labor.

There are always some constraints for solving optimization problems that ignoring them, will made the solutions of the problems unrealistic. In this case, there are constraints such as budget and labor, which are determined by their optimal values in unconstrained case. The values of these constraints are given in Table 4. After solving the problem by considering these constraints, the paths presented in Table 5 are obtained. priGA convergence curve and optimal scheduling of action implementation are shown in Figure 3 and Table 5 respectively.

In this case, for all zones, the implementation of the investment has been result. In addition, investment on covered conductors (action 5) is proposed for zone 4. Investment on ABC cable is not recommended for any zone because of the high capital cost. In the cost analysis, we find that the total cost in this case has increased by 9% due to the constraints. The results of Table 4 shows that the annual cost and labor costs for maintenance and tree trimming did not exceed the defined limits. As is clear from the results of the unconstrained solution, many of the results are more than the set constraints. In the unconstrained solution, we found that due to disregard of constraints, there was little maintenance during the fourth, seventh, and tenth years, but instead, labor and cost density is in other years that may exceed the capabilities and budget and labor capital which shows the importance of considering constraints.

Table 4. Result of optimization problem in 5-year horizon

Year / Yearly Result	→ Constraint	1	2	3	4	5
Maintenance & Failure Cost (K$)	800	754.7	688.55	700.1	600.9	602
Investment (K$)	800	649.4	165.3	8.4	0	0
Man-hour Labor for Maintenance	9000	5572	8760	5074	3418	3983
Man-hour Labor for Tree Trimming	300	227	91	222	188	209
Cost Function (K$)				**3,257.8**		

Table 5. Risk Management action schedules of case study comprising 19 zones in 5-year horizon

year		1	2	3	4	5	year		1	2	3	4	5
z	p						z	p					
1	1	■		■			11	1	■		■	■	
	2	■		■				2	■		■	■	
	3		■					3					■
	4	■						4	■				
	5							5					
	6							6					
2	1	■	■		■	■	12	1	■		■		
	2	■	■		■	■		2	■		■		
	3			■				3		■			
	4	■						4	■				
	5							5					
	6							6					
3	1	■		■	■		13	1	■			■	■
	2	■		■	■			2	■			■	■
	3		■					3		■			
	4	■						4	■				
	5							5					
	6							6					
4	1	■	■			■	14	1	■		■		
	2	■	■			■		2	■		■		
	3				■			3					■
	4	■						4	■				
	5							5					
	6							6					
5	1	■		■	■	■	15	1	■		■		■
	2	■		■	■	■		2	■		■		■
	3		■					3		■			
	4	■						4	■				
	5							5					
	6							6					
6	1			■	■		16	1	■	■		■	
	2			■	■			2	■	■		■	
	3	■	■					3					■
	4							4	■				
	5	■						5					
	6							6					
7	1	■		■	■		17	1		■		■	
	2	■		■	■			2		■		■	
	3		■					3	■				■
	4	■						4					
	5							5	■				
	6							6					
8	1	■		■		■	18	1			■	■	■
	2	■		■		■		2			■	■	■
	3		■					3	■	■			
	4	■						4					
	5							5	■				
	6							6					
9	1		■		■		19	1	■				■
	2		■		■			2	■				■
	3	■		■				3					
	4							4	■				
	5	■						5					
	6							6					
10	1	■	■			■	20	1					
	2	■	■			■		2					
	3			■				3					
	4	■						4					
	5							5					
	6							6					

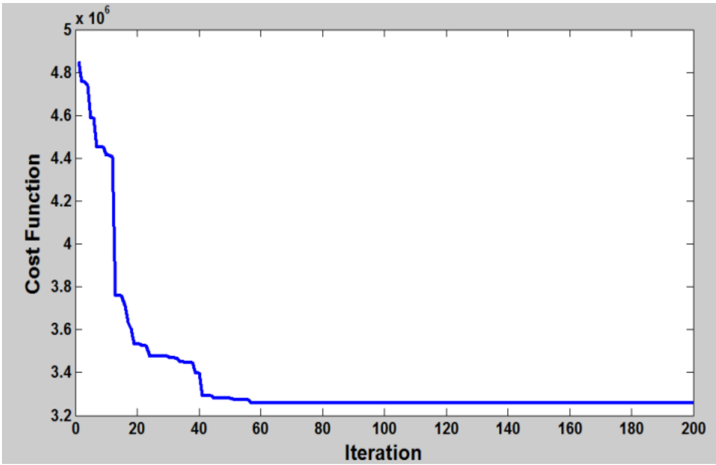


Figure 3. priGA convergence curve for 5-year horizon case

7. CONCLUSION

In this study, strategies have been proposed to improve risk, such as tree trimming, minor and major maintenance, covering pole accessories, replacement of bare conductors with covered conductors and ABCs. The shortest path problem model has been used to implement these strategies in the 5-year horizons. The results of applying the proposed method to the study network indicate that by considering of annual budget, available labor and allocated initial capital, zones with high loads, long repair times, and high failure rates require to replacement of bare conductor by covered conductor and other zones, they need to covering pole accessories. On average for all zones, minor maintenance and tree trimming are also done every other year, that confirming the validity of the modeling. The rate of major maintenance is also obtained on average every four years, which is consistent with the engineering logic of repairs. Overall, the results show the accuracy and precision of the proposed method.

REFERENCES

[1] R. Billinton and J. E. Billinton, "Distribution system reliability indices," *IEEE Transactions on Power Delivery*, vol. 4, no. 1, pp. 561–568, 1989, doi: 10.1109/61.19247.

[2] D. Zhang, W. Li, and X. Xiong, "Overhead line preventive maintenance strategy based on condition monitoring and system reliability assessment," *IEEE Transactions on Power Systems*, vol. 29, no. 4, pp. 1839–1846, 2014, doi: 10.1109/TPWRS.2013.2295379.

[3] L. Bertling, R. Allan, and R. Eriksson, "A reliability-centred asset maintenance method for assessing the impact of maintenance in power distribution systems," *2005 IEEE Power Engineering Society General Meeting*, vol. 3, no. 1, p. 2649, 2005, doi: 10.1109/pes.2005.1489068.

[4] E. Abbasi, M. Fotuhi-Firuzabad, and A. Abiri-Jahromi, "Risk based maintenance optimization of overhead distribution networks utilizing priority based dynamic programming," *2009 IEEE Power and Energy Society General Meeting, PES '09*, pp. 1–11, 2009, doi: 10.1109/PES.2009.5275735.

[5] J. Korpijärvi, "Aging based maintenance and reinvestment scheduling of electric distribution network," Doctor of Science, Institute of Energy Technology, Acta Universitatis Lappeenrantaensis Lappeenranta, Finland, 2012.

[6] R. E. Brown and J. J. Burke, "Managing the risk of performance based rates," *IEEE Transactions on Power Systems*, vol. 15, no. 2, pp. 893–898, 2000, doi: 10.1109/59.867190.

[7] A. Abiri-Jahromi, M. Fotuhi-Firuzabad, and E. Abbasi, "An efficient mixed-integer linear formulation for long-term overhead lines maintenance scheduling in power distribution systems," *IEEE Transactions on Power Delivery*, vol. 24, no. 4, pp. 2043–2053, Oct. 2009, doi: 10.1109/TPWRD.2009.2028785.

[8] A. D. Janjic and D. S. Popovic, "Selective maintenance schedule of distribution networks based on risk management approach," *IEEE Transactions on Power Systems*, vol. 22, no. 2, pp. 597–604, 2007, doi: 10.1109/TPWRS.2007.894863.

[9] P. Dehghanian, M. Fotuhi-Firuzabad, F. Aminifar, and R. Billinton, "A comprehensive scheme for reliability-centered maintenance in power distribution systems—part ii: numerical analysis," *IEEE Transactions on Power Delivery*, vol. 28, no. 2, pp. 771–778, Apr. 2013, doi: 10.1109/TPWRD.2012.2227833.

[10] P. Dehghanian, M. Fotuhi-Firuzabad, F. Aminifar, and R. Billinton, "A comprehensive scheme for reliability centered maintenance in power distribution systems - part i: methodology," *IEEE Transactions on Power Delivery*, vol. 28, no. 2, pp. 761–770, 2013, doi: 10.1109/TPWRD.2012.2227832.

[11] P. Wang and R. Billinton, "Reliability cost/worth assessment of distribution systems incorporating time-varying weather conditions and restoration resources," *IEEE Transactions on Power Delivery*, vol. 17, no. 1, pp. 260–265, 2002, doi: 10.1109/61.974216.

[12] M. Al-Muhaini and G. T. Heydt, "A novel method for evaluating future power distribution system reliability," *IEEE Transactions on Power Systems*, vol. 28, no. 3, pp. 3018–3027, 2013, doi: 10.1109/TPWRS.2012.2230195.

[13] M. Gui, A. Pahwa, and S. Das, "Bayesian network model with Monte Carlo simulations for analysis of animal-related outages in overhead distribution systems," *IEEE Transactions on Power Systems*, vol. 26, no. 3, pp. 1618–1624, 2011, doi:

10.1109/TPWRS.2010.2101619.

[14] L. Goel, X. Liang, and Y. Ou, "Monte Carlo simulation-based customer service reliability assessment," *Electric Power Systems Research*, vol. 49, no. 3, pp. 185–194, Apr. 1999, doi: 10.1016/s0378-7796(98)00121-7.

[15] L. Goel, "Monte Carlo simulation-based reliability studies of a distribution test system," *Electric Power Systems Research*, vol. 54, no. 1, pp. 55–65, Apr. 2000, doi: 10.1016/S0378-7796(99)00066-8.

[16] C. F. Chang, "Reconfiguration and capacitor placement for loss reduction of distribution systems by ant colony search algorithm," *IEEE Transactions on Power Systems*, vol. 23, no. 4, pp. 1747–1755, 2008, doi: 10.1109/TPWRS.2008.2002169.

[17] K. Xie, J. Zhou, and R. Billinton, "Reliability evaluation algorithm for complex medium voltage electrical distribution networks based on the shortest path," *IEE Proceedings: Generation, Transmission and Distribution*, vol. 150, no. 6, pp. 686–690, 2003, doi: 10.1049/ip-gtd:20030797.

[18] M. Gen, R. Cheng, and L. Lin, "Network models and optmization multiobjective genetic algorithm approach," *Network Models and Optmization Multiobjective Genetic Algorithm Approach*, p. 522, 2008.

[19] C. J. Wallnerström, *On incentives affecting risk and asset management of power distribution*. 2011.

[20] K. Sand, O. Gjerde, and D. E. Nordgard, "Current risk exposure in the distribution sector. Initial study," SINTEF Energy Research, 2007.

[21] D. Hughes, G. Dennis, J. Walker, and C. Williamson, "Condition based risk management (CBRM) - Enabling asset condition information to be central to corporate decision making," *Proceedings of the 1st World Congress on Engineering Asset Management, WCEAM 2006*, pp. 1212–1217, 2006, doi: 10.1007/978-1-84628-814-2_133.

[22] J. Schneider *et al.*, "Asset management techniques," *International Journal of Electrical Power and Energy Systems*, vol. 28, no. 9, pp. 643–654, 2006, doi: 10.1016/j.ijepes.2006.03.007.

[23] R. S. Blake, "Methodologies for the evaluation and mitigation of distribution network risk," *Durham University*, 2010.

[24] B. Wareing, "Wood pole overhead lines," *Wood Pole Overhead Lines*, pp. 1–337, 2005, doi: 10.1049/PBPO048E.

[25] R. E. Brown, G. Frimpong, and H. L. Willis, "Failure rate modeling using equipment inspection data," *IEEE Transactions on Power Systems*, vol. 19, no. 2, pp. 782–787, 2004, doi: 10.1109/TPWRS.2004.825824.

[26] B. R. Hunt, R. L. Lipsman, and J. M. Rosenberg, *A Guide to MATLAB®*, vol. 39, no. 06. Cambridge University Press, 2014.

BIOGRAPHIES OF AUTHORS

Adnan Hasan Tawafan was born in Kerbala, Iraq, in 1971. He received the M.S. degree in electrical engineering from University of Technology, Baghdad, Iraq in 2002, and the Ph.D. degree in electrical engineering from UTeM, Malaysia in 2014. Since 2005, he has been with electrical technical department in Technical Institute of Karbala, Iraq. His research interests concern: analysis and design of distribution system, protection, power quality improvement, power electronics applications and risk management. He can be contacted at email: inkr.adn@atu.edu.iq.

Dhafer Mayoof Alshadood was born in Nasiriyah, Iraq, in 1963. He received the B.S. degree in electrical engineering from Basra University, Basra, Iraq in 1987, and the M.S. degree in electrical engineering from Ferdowsi University, Iran in 2018. Currently, he is PHD student in Shahid Chamran University, Iran. He has been with electrical technical department in Nasiriyah Technical Institute, Iraq scine 2010 until 2019. His research interests concern: analysis and design of distribution system, power electronics applications, protection, power quality improvement and risk management. He can be contacted at email: eng.dafer@stu.edu.iq.

Fatima Kadhem Abd was born in Kerbala, Iraq, in 1968. She received the B. degree in electrical engineering from University of technology, Baghdad, Iraq in 1990, Since 2000, she has been with electrical technical department in Technical Institute of Karbala, Iraq. Her research interests concern: analysis of distribution system. She can be contacted at email: fatima@kit.edu.iq.

Improved discrete plant propagation algorithm for solving the traveling salesman problem

Hussein Fouad Almazini[1], Salah Mortada[1], Hassan Fouad Abbas Al-Mazini[1], Hayder Naser Khraibet AL-Behadili[2], Jawad Alkenani[2]
[1]School of Computing, Universiti Utara Malaysia, Kedah, Malaysia
[2]Department of Computer Science, Shatt Alarab University College, Basrah, Iraq

Article Info

Article history:

Received Feb 4, 2021
Revised Oct 14, 2021
Accepted Oct 22, 2021

Keywords:

Genetic algorithm
Metaheuristic
Optimization
Plant intelligence

ABSTRACT

The primary goal of traveling salesman problem (TSP) is for a salesman to visit many cities and return to the starting city via a sequence of potential shortest paths. Subsequently, conventional algorithms are inadequate for large-scale problems; thus, metaheuristic algorithms have been proposed. A recent metaheuristic algorithm that has been implemented to solve TSP is the plant propagation algorithm (PPA), which belongs to the rose family. In this research, this existing PPA is modified to solve TSP. Although PPA is claimed to be successful, it suffers from the slow convergence problem, which significantly impedes its applicability for getting good solution. Therefore, the proposed partial-partitioned greedy algorithm (PPGA) offers crossover and three mutation operations (flip, swap, and slide), which allow local and global search and seem to be wise methods to help PPA in solving the TSP. The PPGA performance is evaluated on 10 separate datasets available in the literature and compared with the original PPA. In terms of distance, the computational results demonstrate that the PPGA outperforms the original PPA in nine datasets which assures that it is 90% better than PPA. PPGA produces good solutions when compared with other algorithms in the literature, where the average execution time reduces by 10.73%.

Corresponding Author:

Hussein Fouad Almazini
School of Computing, Universiti Utara Malaysia
Sintok, Kedah, Malaysia
Email: h.almazni22@gmail.com

1. INTRODUCTION

Combinatorial optimization problems (COPs) are a subset of mathematical optimization problems that are used in different fields, regardless of whether the structure is complex or simple. Various functional problems, such as the traveling salesman problem (TSP), the assembly line balancing problem, the shortest path tree, and the minimum period tree, can be classified as COPs. TSP is commonly used to assess the efficiency of recently designed approaches to COPs and of those relevant to many significant fields, such as engineering, logistics, and transport. TSP is an NP-hard problem following a Hamiltonian cycle with minimal expense [1], [2]. In the TSP principle, the vendor begins from one city, visits all other cities precisely once upon a time, and returns to the beginning city seeking to get a closed tour with lowest expense. The tour expense depends directly on the tour length [3], [4]. Many researchers have suggested solving TSP, with its simple description and very complicated solution. Initially, exact and approximate (heuristic or metaheuristic) approaches were developed to resolve TSP [5], [6]. Exact methods can solve small TSPs optimally. By contrast, heuristic methods are preferable for large TSPs. Moreover, certain greedy, principle-based algorithms may be used to solve TSP. However, conventional approaches lead to exponential time or unsatisfactory quality. To beat these shortcomings, many metaheuristic algorithms in the literature have been developed for TSP due to the importance of accomplishing improved solutions in realistic computing time [7].

Meta-heuristic algorithms can be classified into two major groups; single-based solution and population-based solution [6], [8], [9]. The ability of meta-heuristic algorithms to address optimization problems, such as TSP, relies on two elements: exploitation and exploration. Exploration refers to the research within the search space of unvisited regions, whereas exploitation refers to the search in the existing problem space regions for good solutions [10]–[12]. Single-based algorithms, including variable neighborhood search [13], simulated annealing [14], and guided local search [15], aim to improve a single candidate solution. By contrast, population-based algorithms maintain and enhance candidate solutions, often using population features to conduct direct search; such algorithms include biogeography-based optimization [16], grey wolf optimizer [17], particle swarm optimization (PSO) [18], emperor penguin colony [19], genetic algorithm (GA) [20], ant colony optimization (ACO) [21], black hole (BH) algorithm [22], and dragonfly algorithm (DA) [23]. In recent years, studies on plants have demonstrated that plants display intelligent behavior. Consequently, plants are believed to have a nervous system [24]. Examples of plant intelligence algorithms include the sapling growing up algorithm [25], rooted tree optimization [26], runner root algorithm [27], and strawberry algorithm as plant propagation algorithm (PPA) [28].

PPA was initially suggested by [28] to solve numerical problems; it emulates the survival technique adopted by plants, in which they survive by colonizing new areas with good growing conditions. The strawberry plant has a survival of sustainability and growth that send short runners to exploit the quest for good solutions in existing problem space regions and send long runners in the search space to explore unvisited regions. In [29], studied PPA to solve TSP and showed that PPA can produce better solutions than other algorithms. However, applying a deterministic local search based on 2-opt and k-opt takes exponential time to find an optimal solution; moreover, when this occurs slows down its convergence speed, simply because there may be a deficiency of diversity in certain solutions which leads not to thrust the algorithm towards optimal regions. Consequently, to ensure better convergence, and make the algorithm have solution diversity in both local and global search. This present study implements a crossover operation and three mutation operations (flip, swap, and slide) in PPA and the proposed termed as partial-partitioned greedy algorithm (PPGA). The main contribution of a scientific study is to improve PPA for solving TSP using TSPLIB and produce a promising variant of PPA. The proposed algorithm PPGA is evaluated using 10 TSP datasets (with different sizes and complexities) selected from traveling salesman problem library (TSPLIB). The proposed PPGA is also compared with five metaheuristic algorithms: ACO, PSO, GA, BH, and DA. The main advantage of PPGA is the ability to find an ideal or near-ideal solution in a short time.

This paper is structured being as: section 2 explains the mathematics of TSP. Section 3 provides the literature review. Afterward, section 4 discussed the proposed PPGA methods. Section 5 and 6 explores the experimental results, performance evaluation, and benchmark datasets used in this study are presented. Lastly, section 7 the conclusions and recommendations for potential future research are given.

2. TRAVELING SALESMAN PROBLEM

The importance of TSP is attributed to the detailed studies and high guidelines of computer scientists for it to be included in the assessment of modern optimization algorithms. This problem has been shown to be an NP-hard problem. It can be defined being as: An agent must visit N nodes exactly once and return at the starting node at the lowest expense, i.e., lowest time of visitation or the shortest distance. A cost matrix $C = [c_{ij}]$ is explored to obtain a permutation $\pi : \{0, \dots, N-1\} \rightarrow \{0, \dots, N-1\}$, where c_{ij} shows the expense of visiting node (j) from node (i). The aim is to reduce an objective function represented by f (π, C) as shown in (1):

$$f(\pi, C) = \sum_{i=0}^{N-1} d\ (c_{\pi(i),}\ c_{\pi(i+1)}) + (c_{\pi(N),}\ c_{\pi(1)}) \tag{1}$$

where $\pi(i)$ shows the i^{th} node in permutation π, d is the distance between nodes and $c_{ij}=c_{ji}$ $\forall$ i, j and the position of city (i) can be verified by utilizing the values of the x-axes, and y-axes, i.e., x_i and y_i sequentially.

3. LITERATURE REVIEW

Various algorithms, including single and hybrid algorithms, for TSP have been developed. In [30] proposed a hybrid approach using PSO to improve ACO performance parameters. In addition, a 3-opt local search was endowed to the proposed approach to enhance local search. However, the proposed algorithm has many operations that consume additional time in improving the same search regions to determine the best improvement. In [31] solved the issue of unstable nature instance problem by providing ACO with a local search operator. This operator iteratively chooses the best solution found by the algorithm and then continues to remove or insert cities to improve the solution quality. Nonetheless, using multi-operator in local search may radically increase the computation time or reduce their performance. GA has been implemented by several researchers for TSP. In [32] proposed the use of GA to resolve the challenging large-scale colored balanced TSP. nevertheless, the proposed next generation access (NGA) algorithm should demonstrate good performance in terms of solving speed or solution quality.

In accordance with [33] studied a hybrid metaheuristic algorithms address TSP based on simmulated annealing (SA) and the symbiotic organisms search. The possible challenge of the proposed algorithm can be found to a few considerations, the proposed algorithm includes the use of several parameters. Additionally, increasing the problem complexity, the configuration of the algorithm is linked to it. In another related work [34] studied on improving the performance of the SA by using a greedy search to deal properly with large-scale TSP. However, the proposed algorithm stuck in local optima. This is because the SA utilizes a greedy acceptance criterion that only takes an optimized solution and excludes the worst solution. The probabilistic TSP was solved in multiple trials in [35] through an adaptive multiswarm PSO. In the suggested adaptive PSO, random values are allocated in the initial stage of the search. Next, these parameters are configured dynamically at the same time as the objective function of the problem is optimized. Nonetheless, the search process lacks feedback information and learning due to having less values of parameters to optimize. In [36] strengthened PSO to solve the imprecise cost matrix TSP. The PSO modifications consist of the adoption of the swap series, the swap process, and the guidelines for various speed updates. Nonetheless, several methods have been used in the proposed algorithm which affects to take additional time to get the best improvement. In [37] proposed a hybrid between firefly algorithm (FA) and GA; here, the distance of the FA is redefined by presenting two swap methods to prevent dropping into local optima. The created population which has poor solutions that could lead to long-term convergence to an ideal solution.

According to [38] investigated the BH algorithm to solve TSP. The implementation of the BH algorithm was assessed on 10 datasets and the outcomes in comparison with other optimization techniques. The computational results showed that the BH algorithm can provide solutions better than ACO, GA, and PSO. However, The BH algorithm still lacks the capability to perform high exploration during the update process. Due to a new solution is produced randomly when the previous solution is not improved. Furthermore, a similar study was conducted in [39] to investigate the DA on solving TSP. PPA has been investigated to work on discrete problems, specifically on TSP [29]. The research concerns the usage of the idea of long and short runners in maximum graphs while looking for Hamiltonian cycles. The performance of the PPA algorithm was tested on a traditional dataset and compared with that of PSO SA, GA, and FA. Experimental results were included; however, the performance of the algorithm in solving TSP must be further investigated and compared with that of other optimization methods. Besides, the PPA algorithm suffers from slows down its convergence speed, simply because there may be a deficiency of diversity in certain solutions which leads not to thrust the algorithm towards optimal regions.

4. PROPOSED PARTIAL-PARTITIONED GREEDY ALGORITHM FOR TRAVELING SALESMAN PROBLEM

PPGA randomly begins with the initial population of plants/tours/solutions and iteratively improvises solutions for a given problem instance. In each iteration, PPGA improvises solutions by using short and long runners. The PPGA algorithm proceeds as:

4.1. Initial population

The initial population is a collection of an ordered list of plants where every plant represents a sequence of cities. X_i is tour i, i=1, . . ., NP, implying that NP is the plant population size. In accordance with the Euclidean distance, tour lengths X_L are calculated. Each city of plant is assigned a label of city such that no city can be seen twice in the same plant. TSP tour representation primarily has two strategies: adjacency and path. In this study, path representation is chosen for a tour. As shown in Figure 1, let {A, B, C, D, E} be the labels of cities where A is the starting point.

A	B	C	D	E	A

Figure 1. Plant/tour

4.2. Short runners (exploitation task)

A pre-determined tour set is taken from the ones that had short-length tours after sorting the tours by their tour lengths; from these tours, short runners are sent, i.e., new local tours are produced from them. Crossover is the synthesis of the two tours to create new tours that are copied into the new tours. The crossover operation, which utilizes the crossover simplest case, randomly chooses two tours to crossover, randomly picks a crossover point on the basis of (2), and then changes all cities after that point.

$$CP_{i...n} = \text{r}\,.(L_{i...n} - 2) + 1 \quad (2)$$

where CP shows the crossover point, r ∈ [0, 1] is a randomly selected index, and L is the size of plants.

4.3. Long runners (exploration task)

Long runners are implemented by using three mutation operations (flip, swap, and slide). To explain the flip, swap, and slide operations, A, B, C, D and E are considered cities in the tour visited in sequence {A→B→C→D→E}. The operation structures produced are presented as:

- Flip: Orders the cities vice versa from the last city to the first city; the new sequence is {A → E → D → C → B}.
- Swap: Swaps two cities in the tour, where B and E are exchanged the new sequence is {A→E → C → D → B}.
- Slide: Slides two cities (B and C) after all the other cities' positions; the new sequence is {A→ D → E → B → C}.

PPGA started with a good population of tours (plants). The initial population diversity is supposed to be created by random methods utilized to create tours. Therefore, short runners conduct the exploitation process, and long runners control the exploration process in the search space. Figure 2 shows the pseudo-codes of the PPGA algorithm.

```
PPGA Algorithm
1. Create a population P = X_i, i = 1, . . ., NP of valid tours; select values for g_max and y.
2. g = 1
3. while g < g_max do
4.   Calculate N_i = f (X_i), ∀ X_i ∈ P
5.   Sort N = N_i, i = 1, . . ., NP in ascending order (for minimization);
6.   for i = 1: E (NP/10), Top 10 % of plants do
7.   Create (y/ i) short runners and select two random plants i..i+1 apply crossover
operator, where y
   is an arbitrary parameter.
8.      if N_{i..i+1} > f (r_{i..i+1} ) then
9.     X_{i..i+1}  ← r_{i..i+1}
10     else
11. Ignore r_{i..i+1}
12.      end if
13.  end for
14.  for i = E (NP/10) + 1: NP do
15.   r_i = 1 runner for plant i using random mutation operator, mutation operator = 3, 1 long
runner for each plant not in the top 10 percent
16. if N_i > f (r_i) then
17.      X_i ← r_i
18.  else
19.    Ignore r_i
20. end if
21. end for
22.end while
```

Figure 2. Pseudo-codes of the PPGA algorithm

5. EXPERIMENTAL SETUP

This section presents the performance and robustness of PPGA in solving TSP. Two experimental results are used. The first experiment PPGA against discrete PPA. In the second experiment, the proposed PPGA is compared with five population-based algorithms, i.e., ACO, PSO, GA, BH, and DA. The proposed PPGA is tested on 10 benchmark TSP datasets with diverse characteristics taken from TSPLIB [3]. Choosing different instance structures provides great insights into the behavior of the proposed algorithm when addressing TSP. Figure 3 simplifies one type of TSP instances extracted from the TSPLIB benchmark library.

```
NAME: ulysses22.tsp
TYPE: TSP
COMMENT: Odyssey of Ulysses (Groetschel/Padberg)
DIMENSION: 22
EDGE_WEIGHT_TYPE: GEO
DISPLAY_DATA_TYPE: COORD_DISPLAY
NODE_COORD_SECTION
1 38.24 20.42
2 39.57 26.15
3 40.56 25.32
4 36.26 23.12
5 33.48 10.54
6 37.56 12.19
7 38.42 13.11
8 37.52 20.44
9 41.23 9.10
10 41.17 13.05
11 36.08 -5.21
12 38.47 15.13
13 38.15 15.35
14 37.51 15.17
15 35.49 14.32
16 39.36 19.56
17 38.09 24.36
18 36.09 23.00
19 40.44 13.57
20 40.33 14.15
21 40.37 14.23
22 37.57 22.56
EOF
```

Figure 3. Sample structure of the Ulysses22 dataset

In all datasets, n nodes reflect unique positions in specific towns, e.g., Berlin. The first five lines provide some details, such as the data type (including Euclidean, geographical or other forms of data) about the problem being discussed. The keyword TYPE defines the category of data, e.g., symmetric, asymmetric, or tour set. The keyword DIMENSION is the number of nodes for TSP datasets. The keyword EDGE WEIGHT TYPE determines how the edge weight is described, e.g., the keyword EUC 2D is the Euclidean distance in the plane, whereas GEO is the geographical distance. The node coordinate part starts with the keyword NODE COORD_ SECTION. The node identifier, x and y coordinates are made up of every line. The identifier of the node is a unique integer ≥ 1. Table 1 summarizes the statistics for several TSP instances.

Table 1. Description of some TSP instances

Data name	Location
ulysses22	Groetschel/Padberg
bays29	Groetschel, Juenger, Reinelt
bayg29	Groetschel, Juenger, Reinelt
att48	Padberg/Rinaldi
eil51	Christofides/Eilon
berlin52	Berlin (Germany)
st70	Smith/Thompson
eil76	Christofides/Eilon
gr96	Europe
eil101	Christofides/Eilon

The proposed PPGA algorithm is implemented in the programming environment MATLAB 2020a and executed in an Intel ® Core™ i5 CPU, 2.40 GHz, 4 GB RAM memory, and Windows 7. Each test is conducted of five independent runs. The maximum number of generations (gmax) and population size are set to 200 and 100, respectively, for the proposed PPGA algorithm.

6. EXPERIMENTAL RESULTS

The experimental results of the proposed PPGA is compared with PPA, as shown in Table 2. The best, worst, average, standard deviation (Std), and time (in seconds) are listed for each algorithm. Table 2 compares the results of the proposed PPGA against PPA for ten various datasets TSP. All the problems have been run five times independently.

Table 2. Comparison of PPGA against PPA

Dataset	Algorithm	Best	Worst	Average	Std	Time
ulysses22	PPA	66.81	84.16	73.33	6.92	1.05
	PPGA	75.30	76.58	75.92	0.56	4.86
bays29	PPA	9883.32	10504.77	10257.19	265.89	1.06
	PPGA	9105.87	9764.46	9311.17	287.96	6.44
bayg29	PPA	9253.00	10823.40	10070.52	763.67	1.06
	PPGA	9120.33	9568.70	9318.19	178.09	2.57
att48	PPA	42308.39	47767.24	44816.51	2020.77	1.09
	PPGA	33961.11	34581.00	34585.88	495.81	9.33
eil51	PPA	504.75	604.33	535.98	39.11	1.10
	PPGA	444.03	458.42	450.39	5.20	9.91
berlin52	PPA	8971.38	10420.33	9864.75	537.23	1.11
	PPGA	7748.63	8359.85	8190.14	261.48	10.00
st70	PPA	924.72	1125.70	1045.35	74.88	1.10
	PPGA	714.36	804.27	770.25	34.71	13.14
eil76	PPA	740.36	841.25	786.78	44.17	1.16
	PPGA	586.47	632.15	604.68	17.33	14.35
gr96	PPA	1014.70	1067.18	1038.06	21.55	1.17
	PPGA	634.36	721.80	670.88	34.65	17.92
eil101	PPA	1066.95	1194.35	1137.02	60.34	1.20
	PPGA	774.68	835.36	796.48	27.11	18.85
Average	PPA	7473.43	8443.27	7962.54	383.45	1.11
	PPGA	6316.51	6580.25	6477.39	134.29	10.73

According to the results in Table 2 column 3, the proposed PPGA algorithm achieved nine out of ten datasets, which means 90% better than PPA in terms of best distance. The average comparison was shown in the lower section of the table. Table 2 results indicate that PPGA was better than PPA, according to the average tour costs and the average standard deviation for all datasets shown in column five and six. For the ten selected datasets, the average cost of touring PPGA was 6477.39. The achieved average tour costs for PPA was 7962.54. According to Table 2, this result is due to the improvement process achieved by the crossover and three mutation operations (flip, swap, and slide) which overcome the problem of slows down its convergence speed, and deficiency of diversity in discrete PPA.

Owing to the fact that there are a vast number of articles suggested for TSP in the literature, this study selected the algorithms that were recently published and those that obtained the best outcomes using the same datasets in the experiment to be compared with them in this study. The proposed PPGA is compared with five algorithms that are available in the literature: i.e., ACO, PSO, GA, and BH by [38] and DA proposed by [39]. The computational results are presented in Table 3. The best, worst, average, standard deviation (Std), and time (in seconds) are stated for each algorithm in Table 3.

Table 3 illustrates data showing that PPGA obtains the best results in six datasets out of 10 datasets, which means 60% better than ACO. PPGA outperforms PSO, GA, and DA in all 10 datasets. PPGA also achieves better outcomes in 5 datasets and equal results in one dataset than BH. Small values demonstrate the best solutions obtained and vice versa. In addition, for each algorithm, the Std on various runs is given to demonstrate algorithm efficiency and stability. A description of the average comparison is shown in the lower part of the table. Along with the results given in Table 3, the average tour costs in column five for all the datasets prove that PPGA is better than ACO, PSO, GA, BH, and DA. Based on the results in Table 3, the proposed PPGA algorithm superiority other algorithms in some data and rival in other data this is due to the improvement process achieved in PPGA by the crossover and three mutation operations (flip, swap, and slide). Where crossover and three mutation operations (flip, swap, and slide) have a responsibility in the balance between exploitation and exploration. The average tour cost for PPGA is 6477.39 for the selected ten problems. The accomplished average tour costs for ACO, PSO, GA, BH, and DA are 7089.73, 8307.81, 7661.78, 6546.42, and 6994.93, respectively. Moreover, the standard solution deviation achieved by the PPGA algorithm is slightly worse than DA, but better than ACO, PSO, GA, and BH. This result implies that in seeking optimal solutions, the PPGA algorithm is more efficient and robust, whereas other algorithms such as ACO, PSO, GA, and BH may be stuck in local optimal solutions. Finally, the proposed PPGA algorithm achieved a better average execution time than other optimization algorithms.

Table 3. Comparison of PPGA versus other algorithms

Dataset	Algorithm	Best	Worst	Average	Std	Time
ulysses22	ACO	75.39	75.84	75.48	0.19	84.27
	PSO	75.91	77.18	76.21	0.55	61.87
	GA	75.77	76.44	75.98	1.23	63.39
	BH	75.30	75.93	75.68	0.34	50.44
	DA	76.82	80.93	77.83	1.16	21.00
	PPGA	75.30	76.58	75.92	0.56	4.86
bays29	ACO	9239.19	11014.44	9823.20	722.41	88.25
	PSO	9120.33	9498.17	9195.90	168.97	88.82
	GA	9751.42	10513.91	10015.23	319.87	57.11
	BH	9396.47	9507.17	9463.25	60.95	52.10
	DA	9387.03	9611.78	9480.29	64.96	22.00
	PPGA	9105.87	9764.46	9311.17	287.96	6.44
bayg29	ACO	9447.49	11033.54	9882.21	675.83	99.95
	PSO	9329.25	11332.72	9947.02	799.40	75.29
	GA	9579.12	10411.19	9771.95	127.11	56.16
	BH	9375.44	9375.44	9375.44	0.00	45.87
	DA	9464.41	9704.98	9547.75	64.75	22.00
	PPGA	9120.33	9568.70	9318.19	178.09	2.57
att48	ACO	35230.90	46204.24	39436.18	4874.29	133.45
	PSO	36996.44	61421.99	47018.41	9685.89	84.73
	GA	35312.51	50671.45	43620.63	2004.00	57.35
	BH	34200.86	35528.51	34473.84	589.80	43.21
	DA	37225.85	38683.21	37759.73	425.69	23.00
	PPGA	33961.11	34581.00	34585.88	495.81	9.33
eil51	ACO	454.38	469.05	461.01	6.29	59.19
	PSO	469.15	737.52	574.80	107.23	57.25
	GA	448.83	462.11	453.47	9.41	59.63
	BH	437.89	526.89	458.92	38.63	44.39
	DA	471.56	491.65	475.16	4.51	23.00
	PPGA	444.03	458.42	450.39	5.20	9.91
berlin52	ACO	7757.02	10541.12	8522.90	1152.2	65.07
	PSO	9218.46	14279.43	11089.52	2067.93	68.64
	GA	8779.75	9565.37	9288.44	1301.21	52.73
	BH	8188.07	9356.74	8455.83	508.98	43.40
	DA	9400.75	9610.15	9486.70	72.54	23.00
	PPGA	7748.63	8359.85	8190.14	261.48	10.00
st70	ACO	711.65	855.20	757.75	59.60	94.56
	PSO	1030.84	1756.12	1321.81	269.27	55.28
	GA	1112.30	1242.20	1158.84	52.17	55.09
	BH	723.26	1081.10	797.57	125.22	45.33
	DA	797.47	887.08	839.01	24.28	29.00
	PPGA	714.36	804.27	770.25	34.71	13.14
eil76	ACO	574.24	665.99	594.14	40.21	61.74
	PSO	804.26	1195.90	975.63	152.40	56.76
	GA	619.22	679.78	652.05	122.09	46.69
	BH	566.24	925.84	659.10	152.17	46.54
	DA	624.92	674.48	644.89	13.03	30.00
	PPGA	586.47	632.15	604.68	17.33	14.35
gr96	ACO	555.75	639.91	580.54	33.93	84.38
	PSO	1095.11	1728.82	1378.86	247.50	56.21
	GA	737.96	748.35	742.42	4.32	63.24
	BH	546.83	1197.87	807.24	258.81	43.58
	DA	671.00	836.00	734.05	47.26	40.00
	PPGA	634.36	721.80	670.88	34.65	17.92
eil101	ACO	725.09	868.20	763.92	59.96	89.63
	PSO	1158.70	1973.81	1499.99	319.74	62.09
	GA	828.88	854.43	838.83	9.96	55.18
	BH	720.38	1249.86	897.38	210.14	45.83
	DA	812.80	997.60	898.52	47.90	36.00
	PPGA	774.68	835.36	796.48	27.11	18.85
Average	ACO	6477.11	8236.75	7089.73	762.49	86.04
	PSO	6929.84	10400.17	8307.81	1381.88	66.69
	GA	6724.57	8522.52	7661.78	395.13	56.65
	BH	6423.07	6882.53	6546.42	194.50	46.06
	DA	6893.26	7157.78	6994.93	76.60	26.9
	PPGA	6316.51	6580.25	6477.39	134.29	10.73

7. CONCLUSION

For several decades, the creation of intelligent behavior in plants to solve complex problems has been an important research area. In this study, discrete PPA is improved to solve COPs, particularly the TSP. The algorithm is evaluated on ten benchmark datasets and compared with five common algorithms in the literature to determine the efficacy of the proposed PPGA algorithm for solving TSP. The experimental results indicate that, relative to ACO, PSO, GA, BH, and DA, the PPGA algorithm can yield high-quality solutions. Moreover, the experimental results demonstrate that PPGA considerably outperforms other algorithms in terms of average tour cost and execution time. Further research concerns extending PPGA to solve other COPs such as, the facility layout problem, the quadratic assignment problem, and vehicle routing problems.

REFERENCES

[1] T. Sahai, "Dynamical systems theory and algorithms for NP-hard problems," *Studies in Systems, Decision and Control*, vol. 304, pp. 183–206, 2020, doi: 10.1007/978-3-030-51264-4_8.

[2] S. Mortada and Y. Yusof, "A neighbourhood search for artificial bee colony in vehicle routing problem with time windows," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 3, pp. 255–266, Jun. 2021, doi: 10.22266/ijies2021.0630.22.

[3] G. Reinelt, "TSPLIB. A traveling salesman problem library," *ORSA journal on computing*, vol. 3, no. 4, pp. 376–384, 1991, doi: 10.1287/ijoc.3.4.376.

[4] R. Sagban, K. R. Ku-Mahamud, and M. S. Abu Bakar, "Reactive max-min ant system with recursive local search and its application to TSP and QAP," *Intelligent Automation and Soft Computing*, vol. 23, no. 1, pp. 127–134, 2017, doi: 10.1080/10798587.2016.1177914.

[5] A. Kumar, "Improved genetic algorithm to solve small scale travelling salesman problem," in *Proceedings of the International Conference on Intelligent Computing and Control Systems, ICICCS 2020*, 2020, pp. 516–520, doi: 10.1109/ICICCS48265.2020.9120880.

[6] K. Hussain, M. N. Mohd Salleh, S. Cheng, and Y. Shi, "Metaheuristic research: a comprehensive survey," *Artificial Intelligence Review*, vol. 52, no. 4, pp. 2191–2233, 2019, doi: 10.1007/s10462-017-9605-z.

[7] D. Karaboga and B. Gorkemli, "Solving traveling salesman problem by using combinatorial artificial bee colony algorithms," *International Journal on Artificial Intelligence Tools*, vol. 28, no. 1, 2019, doi: 10.1142/S0218213019500040.

[8] H. N. K. Al-Behadili, K. R. Ku-Mahamud, and R. Sagban, "Genetic-based pruning technique for ant-miner classification algorithm," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 11, no. 1, p. 304, Feb. 2021, doi: 10.18517/ijaseit.11.1.10826.

[9] H. N. K. AL-Behadili, K. R. Ku-Mahamud, and R. Sagban, "Hybrid ant colony optimization and genetic algorithm for rule induction," *Journal of Computer Science*, vol. 16, no. 7, pp. 1019–1028, 2020, doi: 10.3844/JCSSP.2020.1019.1028.

[10] A. M. Jabbar, K. R. Ku-Mahamud, and R. Sagban, "An improved ACS algorithm for data clustering," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 17, no. 3, pp. 1506–1515, 2020, doi: 10.11591/ijeecs.v17.i3.pp1506-1515.

[11] A. Ullah, N. M. Nawi, J. Uddin, S. Baseer, and A. H. Rashed, "Artificial bee colony algorithm used for load balancing in cloud computing: Review," *IAES International Journal of Artificial Intelligence*, vol. 8, no. 2, pp. 156–167, Jun. 2019, doi: 10.11591/ijai.v8.i2.pp156-167.

[12] A. M. Jabbar, K. R. Ku-Mahamud, and R. Sagban, "Improved self-adaptive ACS Algorithm to determine the optimal number of clusters," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 11, no. 3, pp. 1092–1099, 2021, doi: 10.18517/ijaseit.11.3.11723.

[13] N. Mladenović and P. Hansen, "Variable neighborhood search," *Computers and Operations Research*, vol. 24, no. 11, pp. 1097–1100, Nov. 1997, doi: 10.1016/S0305-0548(97)00031-2.

[14] P. J. M. van Laarhoven and E. H. L. Aarts, "Simulated annealing," in *Simulated Annealing: Theory and Applications*, Dordrecht: Springer Netherlands, 1987, pp. 7–15.

[15] C. Voudouris and E. P. K. Tsang, "Guided Local Search," in *Handbook of Metaheuristics*, vol. 1–2, Boston: Kluwer Academic Publishers, 2018, pp. 185–218.

[16] D. Simon, "Biogeography-based optimization," *IEEE transactions on evolutionary computation*, vol. 12, no. 6, pp. 702–713, 2008.

[17] H. Almazini and K. Ku-Mahamud, "Grey wolf optimization parameter control for feature selection in anomaly detection," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 2, pp. 474–483, Apr. 2021, doi: 10.22266/ijies2021.0430.43.

[18] T.-A. N. Abdali, R. Hassan, R. C. Muniyandi, A. H. Mohd Aman, Q. N. Nguyen, and A. S. Al-Khaleefa, "Optimized particle swarm optimization algorithm for the realization of an enhanced energy-aware location-aided routing protocol in MANET," *Information*, vol. 11, no. 11, p. 529, Nov. 2020, doi: 10.3390/info11110529.

[19] S. Harifi, M. Khalilian, J. Mohammadzadeh, and S. Ebrahimnejad, "Emperor penguins colony: a new metaheuristic algorithm for optimization," *Evolutionary Intelligence*, vol. 12, no. 2, pp. 211–226, 2019, doi: 10.1007/s12065-019-00212-x.

[20] M. Mitchell, "An introduction to genetic algorithms," *An Introduction to Genetic Algorithms*, vol. 1996, 2020, doi: 10.7551/mitpress/3927.001.0001.

[21] H. Almazini and K. Ku-Mahamud, "Adaptive technique for feature selection in modified graph clustering-based ant colony optimization," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 3, pp. 332–345, Jun. 2021, doi: 10.22266/ijies2021.0630.28.

[22] A. Hatamlou, "Black hole: a new heuristic optimization approach for data clustering," *Information Sciences*, vol. 222, pp. 175–184, 2013, doi: 10.1016/j.ins.2012.08.023.

[23] S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Computing and Applications*, vol. 27, no. 4, pp. 1053–1073, 2016, doi: 10.1007/s00521-015-1920-1.

[24] S. Akyol and B. Alatas, "Plant intelligence based metaheuristic optimization algorithms," *Artificial Intelligence Review*, vol. 47, no. 4, pp. 417–462, 2017, doi: 10.1007/s10462-016-9486-6.

[25] A. Karci, "Theory of saplings growing up algorithm," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2007, vol. 4431 LNCS, no. PART 1, pp. 450–460, doi: 10.1007/978-3-540-71618-1_50.

[26] Y. Labbi, D. Ben Attous, H. A. Gabbar, B. Mahdad, and A. Zidan, "A new rooted tree optimization algorithm for economic dispatch with valve-point effect," *International Journal of Electrical Power and Energy Systems*, vol. 79, pp. 298–311, 2016, doi: 10.1016/j.ijepes.2016.01.028.

[27] F. Merrikh-Bayat, "The runner-root algorithm: a metaheuristic for solving unimodal and multimodal optimization problems inspired by runners and roots of plants in nature," *Applied Soft Computing Journal*, vol. 33, pp. 292–303, 2015, doi: 10.1016/j.asoc.2015.04.048.

[28] A. Salhi and E. S. Fraga, "Nature-inspired optimisation approaches and the new plant propagation algorithm," in *International Conference on Numerical Analysis and Optimization (ICeMATH)*, 2011.

[29] B. Selamoğlu and A. Salhi, "The plant propagation algorithm for discrete optimisation: The case of the travelling salesman problem," in *Studies in Computational Intelligence*, vol. 637, Springer, 2016, pp. 43–61.

[30] M. Mahi, Ö. K. Baykan, and H. Kodaz, "A new hybrid method based on particle swarm optimization, ant colony optimization and 3-opt algorithms for traveling salesman problem," *Applied Soft Computing Journal*, vol. 30, pp. 484–490, 2015, doi: 10.1016/j.asoc.2015.01.068.

[31] M. Mavrovouniotis, F. M. Muller, and S. Yang, "Ant colony optimization with local search for dynamic traveling salesman problems," *IEEE Transactions on Cybernetics*, vol. 47, no. 7, pp. 1743–1756, 2017, doi: 10.1109/TCYB.2016.2556742.

[32] X. Dong and Y. Cai, "A novel genetic algorithm for large scale colored balanced traveling salesman problem," *Future Generation Computer Systems*, vol. 95, pp. 727–742, 2019, doi: 10.1016/j.future.2018.12.065.

[33] A. E. S. Ezugwu, A. O. Adewumi, and M. E. Frîncu, "Simulated annealing based symbiotic organisms search optimization algorithm for traveling salesman problem," *Expert Systems with Applications*, vol. 77, pp. 189–210, 2017, doi: 10.1016/j.eswa.2017.01.053.

[34] X. Wu and D. Gao, "A study on greedy search to improve simulated annealing for large-scale traveling salesman problem," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2017, vol. 10386 LNCS, pp. 250–257, doi: 10.1007/978-3-319-61833-3_26.

[35] Y. Marinakis, M. Marinaki, and A. Migdalas, "Adaptive tunning of all parameters in a multi-swarm particle swarm optimization algorithm: An application to the probabilistic traveling salesman problem," *Springer Proceedings in Mathematics and Statistics*, vol. 130, pp. 187–207, 2015, doi: 10.1007/978-3-319-18567-5_10.

[36] I. Khan, S. Pal, and M. K. Maiti, "A modified particle swarm optimization algorithm for solving traveling salesman problem with imprecise cost matrix," in *Proceedings of the 4th IEEE International Conference on Recent Advances in Information Technology, RAIT 2018*, 2018, pp. 1–8, doi: 10.1109/RAIT.2018.8389060.

[37] L. Teng and H. Li, "Modified discrete firefly algorithm combining genetic algorithm for traveling salesman problem," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 16, no. 1, pp. 424–431, 2018, doi: 10.12928/TELKOMNIKA.V16I1.4752.

[38] A. Hatamlou, "Solving travelling salesman problem using black hole algorithm," *Soft Computing*, vol. 22, no. 24, pp. 8167–8175, 2018, doi: 10.1007/s00500-017-2760-y.

[39] A. I. Hammouri, E. T. A. Samra, M. A. Al-Betar, R. M. Khalil, Z. Alasmer, and M. Kanan, "A dragonfly algorithm for solving traveling salesman problem," in *Proceedings - 8th IEEE International Conference on Control System, Computing and Engineering, ICCSCE 2018*, 2019, pp. 136–141, doi: 10.1109/ICCSCE.2018.8684963.

BIOGRAPHIES OF AUTHORS

Hussein Fouad Almazini is a Ph.D student at School of Computing, Sintok, Universiti Utara Malaysia, Kedah, Malaysia. He holds a master's degree in information technology in the area (Artificial Intelligence). He works in a artificial intelligence, evolutionary computation, meta-heuristic algorithms and swarm intelligence. He is also interested in mining scientific datasets in domains such as IoT, healthcare, cyber security, social network, business, demographic, sustainability, and intelligence analysis. He can be contacted at email: h.almazni22@gmail.com.

Salah Mortada is a Ph.D student at School of Computing, Sintok, Universiti Utara Malaysia, Kedah, Malaysia. He holds a master's degree in information technology in the area (Usability and User Experience). His current research interests include artificial intelligence, evolutionary computation, meta-heuristic algorithms, swarm intelligence, and their applications in the design and optimization of intelligent transportation management such as VRPs. He can be contacted at email: sms099@yahoo.com.

Hassan Fouad Almazini is a Ph.D. student at School of Computing, Sintok, Universiti Utara Malaysia, Kedah, Malaysia. He holds a master's degree in information technology in the area (Artificial Intelligence). Hassan research and development experience includes over 8 years in the academia and Industry. He works in a artificial intelligence, evolutionary computation, meta-heuristic algorithms and swarm intelligence. He is also interested in mining scientific datasets in domains such as IoT, healthcare, cyber security, social network, business, demographic, sustainability, and intelligence analysis. He can be contacted at email: abba@ahsgs.uum.edu.my.

Dr. Hayder Naser Khraibet Al-Behadili has a Ph.D. degree in Information Technology specialization in artificial intelligence. His research is mainly focused on development of artificial intelligence and data mining techniques based on swarm intelligence algorithms. He can be contacted at email: hayderkhraibet@sa-uc.edu.iq.

Jawad Kadum Hussein is a master student in Information System specialization in data mining. His research is mainly focused on improvement the data mining based on artificial intelligence, evolutionary computation, meta-heuristic algorithms, and swarm intelligence. He can be contacted at email: Jawadalkenani@sa-us.edu.iq.

IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 11, No. 1, March 2022, pp. 23~33
ISSN: 2252-8938, DOI: 10.11591/ijai.v11.i1.pp23-33

Design of the use of chatbot as a virtual assistant in banking services in Indonesia

Bhakti Prabandyo Wicaksono, Amalia Zahra
Computer Science Department, Binus Graduate Program - Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia

Article Info

Article history:

Received Aug 31, 2021
Revised Sep 27, 2021
Accepted Oct 12, 2021

Keywords:

Artificial intelligence
Botika
Chatbot
Rasa

ABSTRACT

A chatbot is a computer program designed to simulate an interactive communication to user (human) via text, audio, or video. Currently several banks in Indonesia have adopted chat technology in customer service. The application of artificial intelligence in customer service aims to prepare banks for the challenges of industry banking 4.0. In addition, it is also to solve problems currently faced by customer service. Implementing chatbot platform in banking in Indonesia is not just plug and play, although there are quite a lot of chatbot platforms available, including Rasa Platform, Botika Platform, and Kata.ai Platform. However, this study only evaluates two chatbot platforms, namely Rasa and Botika, where the two platforms are considered not yet able to be immediately adopted by banks. This is because the application of banking technology in Indonesia must refer to regulatory regulations, including those related to environmental needs, language, speed, and accuracy to understand the intent of users. Hence, research is needed to decide which chatbot platform can be implemented in the banking industry without violating regulatory regulations. From the results of evaluations conducted using the usability and hedonic motivation system adoption system (HMSAM) methods, it is found that users prefer Botika platform to be implemented in the banking industry.

Corresponding Author:

Amalia Zahra
Department Computer Science, Binus Graduate Program - Master of Computer Science, Bina Nusantara University
Jakarta, Indonesia, 11480
Email: amalia.zahra@binus.edu

1. INTRODUCTION

Qwerty Bank is a limited liability company engaged in the financial services industry. Qwerty Bank focuses on the micro, small and medium enterprises (MSME) segment. Currently, it has grown and developed into a bank that has entered the group of medium-sized banks in Indonesia in terms of assets. Along with opening up opportunities and increasing the ability to serve the needs of a wider community, Qwerty Bank has expanded its business to the commercial and consumer segments. These three segments are the business pillars of Qwerty Bank, with services supported by an optimal fund management system, reliable information technology, competence in human resources and good corporate governance practices. This foundation allowed Qwerty Bank to step forward and position it as a credible bank. The operation of Qwerty Bank is currently supported by more than 425 outlets spread across 22 provinces throughout Indonesia which are connected in real time. Qwerty Bank has also built a micro-banking network, which now totals 543 outlets, as part of a partnership program with cooperatives and microfinance institutions.

Customer satisfaction and trust in using bank products and services will be able to attract the interest of prospective new customers [1]. Optimal service certainly provides a positive image for the company thereby increasing the company's image in the eyes of customers and prospective customers. One of the roles is done by officers who deal directly with customers, including marketing, customer service and call centers. Functionally, customer service and call center have the same role, namely as a company representative to interact with customers. Broadly speaking, the difference can be seen from the media used to interact, where customer service uses face-to-face media, while the call center uses telephone media.

One of the services provided by Qwerty Bank to customers is call center service. About 50% of the top 18 customer services are information services. Call Center services have 181 service categories that vary in both banking and credit cards. The services used at Qwerty Bank are using Halo Qwerty 14555 which is managed by a third party where the bank has to pay for such a service based on a fixed scheme every time the number of calls is less than 150,000. However, if there are more than 150,000 calls per month, the cost will be 2,750 per call.

There is a trend of decreasing number of traffic calls from year to year, but this is not directly proportional to the costs incurred for the Halo Qwerty call center service 14555 every year. One of them can be seen in 2018, the average number of call center services has decreased significantly. When viewed in terms of service, this can be interpreted to be a positive thing for the company, because it indicates the reduction in the number of customer complaints against Qwerty Bank services. However, this is inversely proportional in terms of cost. With an average of 115,776 calls per month, the costs incurred for the call center facility every month are Rp 625,000,000.00, or in a year around Rp 7,500,000,000.00. When seen the cost per call, it will increasingly look more expensive every year when compared to the cost per call Rp 2,750.00 if the number of calls reaches 150,000 every month. For example, in 2017, the cost per call is Rp 2,750.00, while in 2018 the cost per call will be Rp 5,389.35.

With the increasing use of Chatbot technology, more and more Chatbot platforms are being developed. This can be shown by the large number of information technology (IT) companies that focus their business on the field of Chatbot, such as Kata.ai (https://kata.ai), Rasa Platform (https://rasa.com), Botika (https://botika.online), and so on. However, there is not yet one Chatbot platform that can be adopted and implemented directly by banks in Indonesia. This is due to the fact that the implementation of information technology in the banking industry in Indonesia is required to refer to the regulations issued by the regulator, in this case Bank Indonesia (BI) and the financial services authority or known as Otoritas Jasa Keuangan (OJK). Some of the regulations issued by the regulator include the existence of on-premise obligations. On-Premise simply means that it must manage the infrastructure independently, both hardware, software, network, and so forth. In addition, the regulator also requires that the software used is the property of the Bank and complemented with at least two-factor authentication. Two-factor authentication is a security process where the user provides two means of identification, one of which is usually a physical sign, like a card, and the other is usually something memorized, such as a security code [2]. These things are challenge for banks in Indonesia for adopting Chatbot technology. These challenges are due to the fact that the existing Chatbot platforms can barely be implemented on-premise basis. In addition, most of several existing Chatbot platforms still use English as a knowledge base, thus the language used by the Chatbot platform is still one of the obstacles encountered considering that Indonesian is still the main language of communication in banking in Indonesia.

Based on the explanation above, there are several services in banking that can be replaced by adopting Chatbot technology. This paper outlines a discussion related to the Chatbot platform that can meet the requirements of banking regulators in Indonesia. It can also use Indonesian as a knowledge base, and how the architecture needs to be adjusted to suit the process business and regulations in the bank industry in Indonesia. Additionally, a discussion on the evaluation of Chatbot platforms will also be elaborated and analyzed further.

2. RESEARCH METHOD

In a study, a method is needed. This method is used as a reference to get the expected results. The method used to complete this paper is shown in Figure 1.

2.1. Methodological approach

A methodological approach is an approach that will be taken to explore the topic. One methodological approach that is commonly used is a literature review. This phase is divided into 2 stages.

2.1.1. Literature review

In the literature study stage, it is carried out by collecting and reading writings in the form of articles and journals related to the research topic. Besides that, it can also be done by discussing with subject matter expert. As for the things that are learned including government regulation and Chatbot.

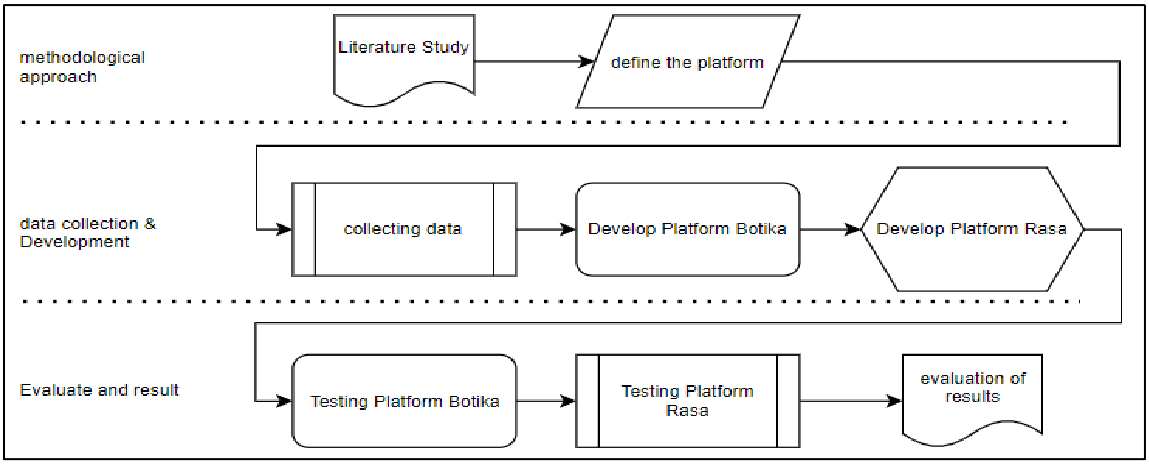


Figure 1. Research method

a. Government regulation

Currently, banks in Indonesia, in carrying out all their activities, must comply with all regulations issued by regulators, namely Bank Indonesia (BI) and the financial services authority Otoritas Jasa Keuangan (OJK). The two regulators, in drafting rules for banking, must refer to other rules defined by the Government of Indonesia. The most recent one is the rules regarding Electronic System and Transaction Management stated in the Government Regulation of the Republic of Indonesia peraturan pemerintah (PP) no. 71/2019. This PP has generally allowed private institutions to use cloud technology. However, it has not been specifically explained what data is allowed.

The latest regulatory rules for the implementation of electronic transactions and systems and digital services in the banking industry are stated in the financial services authority regulation Peraturan Otoritas Jasa Keuangan (POJK) Number 12/POJK.03/2018 concerning the implementation of digital banking services, where the POJK does not refer to PP No. 71 of 2019. Broadly speaking, the POJK still requires every application of digital banking service to use the bank's software with the bank's hardware or the customer's or prospective customer's hardware. Unlike the case with Bank Indonesia Regulation Peraturan Bank Indonesia (PBI) Number 18/40/PBI/2016 concerning the implementation of payment transaction processing. In PBI, every application to be used by a customer or prospective customer must be registered and approved by Bank Indonesia, where in application registration or customer service, it is equipped with several supporting documents, including testing documents (SIT, UAT, and security testing).

b. Chatbot

Chatbot is a computer program designed to simulate an interactive conversation or communication with the user through text, voice, and visuals. Conversations that occur between computers and humans are a form of response from programs that have been declared in the program database on the computer. The resulting response is the result of scanning keywords in words or sentences from the user and produces the response that is considered the most suitable, or the pattern of words that are considered the closest from the database [3]-[12]. In everyday language, Chatbot is an application or computer program designed to imitate humans themselves. The limitation taken from Chatbot is that it can imitate human communication. Thus, if humans are chatting with the program, it is as if there are two humans communicating with each other. In fact, humans communicate with robots. The robot has been designed to respond to all kinds of questions and statements entered by users. This occurs because it has previously been declared in the database, in the form of word entities, sentence patterns, and various types of statements and questions [13]-[15]. The Chatbot component diagram is shown in Figure 2 [16].

The Chatbot software package consists of three components. The components are being as:

– Responder. The responder is the interface that connects with the user and controls the input made by the user and the output of the replies. Data from responder will be sent to Classifier.
– Classifier. Classifier functions to normalize and filter data sent from users. The data submitted by the user will then be replaced and divided into logical components. The classifier moves normalized sentences into the Graph-master component of the Chatbot. The classifier processes the output of the graph-master component. Apart from that, the classifier also handles database syntax instructions.

– Graph Master. The Chatbot graph-master component handles matching patterns. Graph-master is responsible for managing content storage or it can be said that Graph-master is the brain of Chatbot. The graph-master also handles the pattern matching process and pattern matching algorithms.

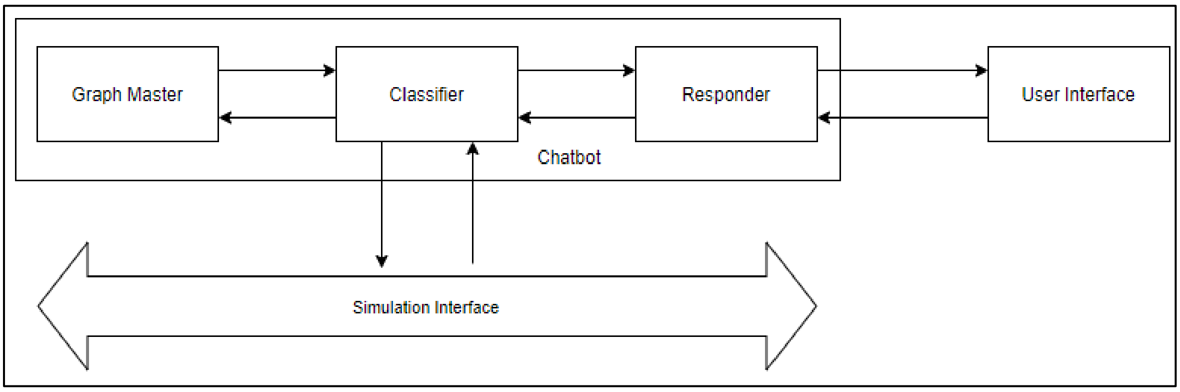


Figure 2. Chatbot component

2.2. Development

The development stage is a stage carried out by researchers to explore existing platforms. In this phase, the necessary data is collected and then the data is used to develop applications. There are three stages in this phase, these stages are being as.

2.2.1. Development of platform botika

Botika is a startup with a chatbot product or a smart chat assistant that has the task for answering user questions and complaints through the Facebook Messenger, LINE and Telegram chat applications. All messages that come through chat applications connected to Botika can be answered from one place. In addition, Botika can reply to messages for 24 hours using the Indonesian language and provide complete information about the products being sold. Hence, the role of Botika is to replace the role of customer service who has limited time to reply to messages from buyers. Besides making work easier, Botika can also reduce company costs for customer service.

The Botika Platform is a chatbot platform whose environment is in a cloud, so that all conversation data that occurs on the chatbot will later be stored in the cloud for data analytic needs. This is one of the obstacles when implementing the Botika Platform in the banking industry in Indonesia, because the current regulatory regulations do not yet clearly state what data can be stored in the cloud. Develop a Chatbot application on the Botika Platform, user access to a dashboard that can be accessed via https://botika.online is required. After successfully logging into the dashboard, the next steps are creating the following aspects:

– State. State is a setting that defines the position of a Chatbot. Thus, the Chatbot will know after which state it will go to which state.
– Action. Action is a command that is executed by Chatbot in any predefined state where to add commands using the PHP programming language.
– Transition. Transition is used to change the condition of a state with the aim that Chatbot can process or move to another state.
– Integration. The Integration feature functions to integrate the bot into a social media or a web platform to be paired. Botika Platform already provides a feature to facilitate Chatbot integration to the Chat Platform on social media, for example WhatsApp, Facebook, Telegram, and so on. However, this research is only integrated into the webchat which is integrated into the Qwerty Bank corporate web.
– Intent. The Intent feature on the Chatbot functions as a knowledge base, meaning that this feature is used to register questions along with their answers. To register, we can use the bulk method (csv file format) or one by one.
– Train intent. Train intent is a feature on the Botika dashboard. This feature is used to carry out the training data process, after an intent is added, so that it can be recognized by the Bot.
– Test Intent. This feature is used to assure that the data being trained can be recognized by the Chatbot through a string similarity approach. The probability level of a word or sentence delivered can be

measured from the Test Intents feature. For example, a user submits the word "Halao". The results can be seen in the Figure 3.

In order for the Botika Platform to comply with regulatory regulations, additional development is required on top of the Botika dashboard. This is carried out for features where confidential data is needed, such as card number, identity number, and date of birth. This additional development is carried out by developing a web-based form created using PHP to send the results to a Qwerty Bank server. Afterwards, the Botika Platform will display a link to a form that will open once a customer clicks it. The form can be filled by customers or prospective customers. That way, confidential data will not be sent and stored on the Botika Platform but stored on servers in the Qwerty Bank Data Center.

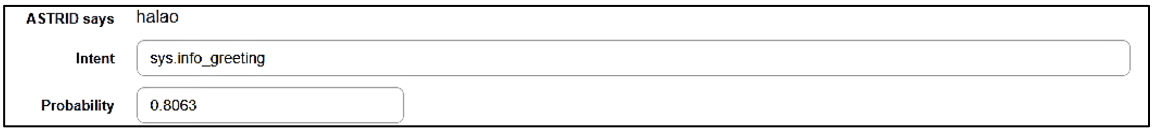


Figure 3. Probability in Botika platform

2.2.2. Development of rasa platform

Rasa Core and Rasa natural language understanding (NLU) is an open sources Python library for making conversational software, with the aim of creating machine learning and language understanding based dialogue management that can be developed by non-specialist software developers. Rasa can be developed by environment on premise and using the Python programming language [17]-[23]. The Rasa Platform can be implemented in an on-premises environment but cannot yet be implemented in the banking industry in Indonesia, because the library currently used is spacy version 1.8.2, where this version cannot understand Indonesian perfectly. This is an obstacle when implementing the Rasa Platform. Develop a Chatbot application on the Rasa Platform, several supporting software are needed, including Anaconda3, Python3, Rasa Package version 1.8.1, and Apache. Such software is installed in one environment. The environment used has the specifications of 1 Core, 4 GB RAM, and 100 GB hard disk drive (HDD). After the supporting software are installed, the next steps are:

– Intent. Intent is a question mapping that will be asked to the created a chatbot. The intent will later become the knowledge base for the chatbot application. To describe the intent on the Rasa Platform, it is saved in a file with .yml extension.
– Config. Config is the process of configuring the components needed by the Chatbot. Each configuration is defined in a file and saved in .yml format.
– Action. Action is a file containing programming in Python, where the file is saved in .py format. The action file is a collection of commands that will be executed by Chatbot, where the commands are based on the variables that have been registered in the domain file. The command can be in the form of response text or connect to a service or database.

When viewed from the regulatory perspective, there is no rule violated when implementing the Rasa Platform in the banking industry in Indonesia, because there is no regulation from the regulator that requires applications to use Indonesian. However, the implementation of chatbot using the Rasa Platform is not quite right due to the fact that the majority of the Qwerty Bank's customers are Indonesians, thus it would be more appropriate if the chatbot also uses Indonesian. For this reason, a modification to the Rasa Platform is carried out by replacing or adding relevant Python libraries, for example by adding the natural language toolkit (NLTK) library which is used for natural language processing such as classification, tokenization, stemming, tagging, and parsing. Moreover, a newer version (at least version 2.0) of the Spacy library is used since it already supports Indonesian.

2.3. Evaluate and result

In evaluate and result stage, a series of tests were carried out on the system that had been developed previously. This is performed to detect errors early, ensure the system's ability to run, and validate the correctness of the system. After testing is complete, the next step is to evaluate the results.

2.3.1. Testing platform botika

This test is carried out using the Blackbox testing method. Blackbox testing is a software testing method that examines application functionality without peering into its internal structure or workings. The results of these tests can be seen in Table 1.

Table 1. Test result from Botika Platform

Feature	Test Scenario	Step	Data	Expected Results	Result	Pass/Fail
Account Balance	Display Balance	1. Greeting 2. Choose "Informasi Saldo" 3. Click Form 4. Input Data (Debit Card Number, Date of Birth) 5. Click Submit 6. Input OTP 7. Click Submit	Card Number: 529******4653 Date of Birth: 2*/0*/1**2	Balance Information is Displayed	Success	Pass
Saving	Show Product Information	1. Greeting 2. Select saving information 3. Choose "Tabungan" 4. Savings information: a. Feature and cost b. Facility c. Benefit	-	Information on Savings Products Appears Successfully	Success	Pass

2.3.2. Testing rasa platform

Blackbox testing is also used to perform testing on the Rasa platform. This is done so that the comparison results are comparable to the previous platform. For the Rasa platform test, the test results are as shown in Table 2.

Table 2. Rest result from Rasa Platform

Feature	Test Scenario	Step	Data	Expected Results	Result	Pass/Fail
Account Balance	Display balance	1. Greeting 2. Type "Informasi Saldo" 3. Type Account Number	Account Number: 4301*****987	Balance Information is Displayed	Success	Pass
Saving	Show product information	1. Greeting 2. Type "Informasi Produk" 3. Choose "Tabungan" 4. Savings information: a. Feature and cost b. Facility c. Benefit	-	Information on Savings Products Appears Successfully	Success	Pass

2.3.3. Evaluation of result

The evaluation was carried out using the usability method and the hedonic motivation system adoption model (HMSAM) method. Both methods are used to obtain a value, where the value is obtained by giving a questionnaire to at least 15 correspondents with the aim that the results obtained were in accordance with predetermined goals and objectives. Usability consists of three components:

- Think-aloud evaluation (TA). Users are asked to have their opinion and express their feelings when interacting with the application.
- Cognitive walkthrough (CW). The evaluator works through series of task scenarios and asks a number of questions from the user's perspective.
- Heuristic evaluation (HE) is a principle used to evaluate the interaction and interface of a system or application [24], [25].

The HMSAM method is a model for measuring a system that adapts to hedonic motivation [26]. Hedonic motivation refers to the influence of a person's pleasure and pain receptors on their willingness to move towards a goal or away from a threat[4]. There are five factors that are the focus of measurement in HMSAM, namely perceived usefulness, perceived ease of use, curiosity, control, and joy, where these five factors will influence behavioral intention to use and immersion of an application [26]. Thus, HMSAM has five aspects that affect the evaluation results. These aspects are:

- Perceived usefulness, used to measure the increase in performance when using a system.
- Perceived ease of use, used to measure the ease of use of the system.
- Curiosity, used to measure the extent to which a system can increase curiosity in cognitive aspects.
- Control, used to measure the user's perception that he is being invited to interact by the system.
- Joy, used to explore the pleasure aspects of users interacting with the system.

To get the values of each usability and HMSAM component, it is performed by giving a questionnaire to at least 15 correspondents. The questionnaire contains 17 questions. The composition is presented in Table 3. Correspondents were asked for their opinion regarding the Chatbot application by providing a choice of categories Strongly Agree (4), Agree (3), Disagree (2), and Strongly Disagree (1) for each question.

Table 3. Composition question usability method and HMSAM method

Component	Method	Quantity
Think-Aloud Evaluation (TA)	Usability	5
Cognitive Walkthrough (CW)	Usability	6
Heuristic Evaluation (HE)	Usability	6
Perceived ease of use	HMSAM	4
Perceived usefulness	HMSAM	3
Curiosity	HMSAM	4
Control	HMSAM	4
Joy	HMSAM	3

3. RESULTS AND DISCUSSION

Based on the methodology previously described, this section discusses the results achieved. In terms of application, it produces two Chatbot applications with different platforms. To make the comparison of the two chatbot platforms more comprehensive, assessments obtained from the user's side were also conducted. Two methods were used: usability and the hedonic motivation system adoption model (HMSAM) method. To get the value of each component, a questionnaire method was used. The questionnaire consisted of 17 statements which were asked to 30 respondents. The 17 statements cover the components of the usability method and the HMSAM. Table 4 describes the results of 30 respondents who filled out the questionnaire.

Table 4. Questionnaire results of usability and HMSAM method

No	Component of Usability	Component of HMSAM	Botika		Rasa	
			Total	Average	Total	Average
1	HE	Perceived ease of use	104	3.467	93	3.1
2	TA	Perceived ease of use	99	3.3	93	3.1
3	HE	Control	97	3.233	88	2.933
4	HE	Joy	92	3.067	83	2.767
5	CW	Perceived ease of use	101	3.367	93	3.1
6	CW	Curiosity	99	3.3	88	2.933
7	TA	Perceived ease of use	98	3.267	92	3.067
8	HE	Control	93	3.1	85	2.833
9	CW	Curiosity	107	3.567	99	3.3
10	CW	Joy	100	3.333	90	3
11	HE	Perceived usefulness	92	3.067	85	2.833
12	TA	Perceived usefulness	102	3.4	94	3.133
13	TA	Perceived usefulness	102	3.4	96	3.2
14	HE	Control	96	3.2	92	3.067
15	TA	Joy	92	3.067	89	2.967
16	CW	Control	89	2.967	83	2.767
17	CW	Curiosity	83	2.767	79	2.633

3.1. Validity and reliability test of the questionnaire

Whether a research instrument is good or not is determined by its validity and reliability. The validity of the instrument concerns the extent of measurement accuracy in measuring what you want to measure, while reliability questions the extent to which a measurement can be trusted because of its consistency. The instrument is said to be valid if it can reveal data from the variables appropriately not deviating from the actual situation. The instrument is said to be reliable if it can reveal reliable data [27]. In this study, the validity and reliability tests were carried out with the results showing that the T table value (5%, 30) was 1.7. The data is declared valid because the value of T count 17 statements is greater than the value of T table. To perform a reliability test, the variance value, the number of variance and the total variance are needed. The variants are obtained from the var function for all respondents' answers in each statement. The total variance is obtained from the sum of all variance values. After obtaining these three values, then it is used to calculate the reliability value obtained by a value of 0.91.

3.2. Usability method

The results in Table 4 are then grouped by components, which are presented in Table 5. The Total value is obtained from the sum of the average values in Table 4 according to the components, where the results of the total column will be divided by the number of questions according to the components, the results of the division are placed in the Average column. From Table 5, user ratings of the two Chatbot platforms prefer the Botika platform to the Rasa platform. This can be proven from the values in the Average and Percentage (%) columns of the two platforms tested. For the TA component, users rated the Botika platform with a score of 82.15% better than the Rasa platform which had a value of 77.325%. Furthermore, for the CW component, the Botika platform has a value of 80.4% while the Rasa platform has a value of 78.875%. Finally, for the HE component, users also rated the Botika platform higher with a value of 79.7% when compared to the Rasa platform which was valued at 73.05%.

Table 5. Result comparison of usability method

Component of Usability	Botika			Rasa		
	Total	Average	Percentage (%)	Total	Average	Percentage (%)
TA	16.433	3.286	82.15	15.466	3.093	77.325
CW	19.3	3.216	80.4	17.733	2.955	73.875
HE	19.133	3.188	79.7	17.533	2.922	73.05

3.3. Hedonic motivation system adoption model (HMSAM)

The results in Table 4 are then grouped by components, which are presented in Table 6. From Table 6, it can be seen that users judge that the Botika platform has a higher value than the Rasa platform. It can be seen from Table 7 that all the HMSAM components, the Botika platform have a higher value. The Perceived ease of use component of the Botika platform is valued at 83.75%, while the Rasa platform has a value of 77.275%. Furthermore, the perceived usefulness component is valued at 82.2% for the Botika platform and 76,275 for the Rasa platform. Then, respectively, the curiosity, control, and joy components on the Botika platform compared to the Rasa platform were 80.275% compared to 73.875%, 78.125% with 72.5%, and 78.875% compared to 72.775%. From Table 6, arithmetic calculations are then carried out to get the behavioral intention to use and immersion values as can be seen in Table 7.

Table 6. Result Comparison of HMSAM Method

HMSAM Component	Botika			Rasa		
	Total	Average	Percentage (%)	Total	Average	Percentage (%)
Perceived Ease of Use	13.4	3.35	83.75	12.366	3.091	77.275
Perceived Usefulness	9.866	3.288	82.2	9.166	3.055	76.375
Curiosity	9.633	3.211	80.275	8.866	2.955	73.875
Control	12.5	3.125	78.125	11.6	2.9	72.5
Joy	9.366	3.155	78.875	8.733	2.911	72.775

Table 7. HMSAM behavioral invention to use dan immersion

	Botika					Rasa				
	Ease to use	Usefulness	Curiosity	Joy	Control	Ease to use	Usefulness	Curiosity	Joy	Control
	3.35	3.288	3.211	3.155	3.125	3.091	3.055	2.955	2.911	2.9
Add Perceived ease to use		6.638	6.561	6.505	6.475		6.146	6.046	6.002	5.991
Average Perceived ease to use		3.319	3.281	3.253	3.238		3.073	3.023	3.001	2.996
							Behavioral intention to use			
Total		9.852					9.097			
Average Behavioral		3.284					3.032			
Percentage Behavioral		82.1					75.808			
							Immersion			
Total		9.771					9.020			
Average Immersion		3.257					3.007			
Percentage Immersion		81.421					75.163			

Table 7 shows that the percentage value of Behavioral intention to use is 82.1% for the Botika Platform and 75.808% for the Rasa Platform. This value is influenced by the perceived usefulness, curiosity, and joy values. As for the Immersion value, the Botika Platform is higher with a value of 81.421% compared to the Rasa Platform which is 75.163%. Immersion value is greatly influenced by the curiosity, joy, and control components. By looking at the value of the Behavioral intention to use and Immersion components, it means that users will use Chatbot technology more when the Botika Platform is implemented rather than the Rasa Platform which is implemented. The data in Table 8 is then used to measure the significance test, namely "t-Test: two-sample assuming equal variances". The results are shown in Table 9.

a. Hypothesis
- H0: µ1 = µ2 (There is no significant difference between the mean Botika and Rasa evaluations).
- H1: µ1> µ2 (There is a significant difference between the mean Botika and Rasa evaluations).

b. Significance Level (a=5% or 0.05).

c. Acceptance criteria
Accept H0 if t count=t table or p-value> alpha (a) or Reject H0 If t count> t table or p-value=alpha (a).

d. Interpretation of the results of statistical analysis hypothesis test t-test
- Mean is the average value of Botika Output=3.239 and the average value of Rasa Output=2.993.
- Variance is the value of the variation in Botika Output = 0.0054 and the value of the variation in the Rasa Output=0.0046.
- Observations are the number of Botika and Rasa observations, each of which consists of 10 observations.
- Pooled Variance is a combined variation of Botika and Rasa, namely 0.0050.
- Hypothesized Mean Difference is the average difference between Botika and Rasa, but for this case example, it is assumed that there is no difference so that the value is "0".
- df is the Degree of Freedom or degrees of freedom obtained by calculating n1+n2-2 so that 8+8-2=14.
- t-stat is the value of t count which is equal to 6.951.
- t critical one tail is t table value that is 1.761.
- Because the hypothesis shows one direction, namely µ1>µ2 (bigger), then what is seen is only the p-value and t table (t critical) in one direction, namely ONE TAIL.

e. Statistical conclusions
- t count (6.951)>t Table (1.761) means we reject H0 (ACCEPT H1), or
- p-value (3.3743)>alpha (0.05) means we reject H0 (ACCEPT H1).
- Botika is more suitable to be implemented based on the significant differences with the Rasa.

Table 8. Average component HMSAM

No	HMSAM Component	Botika	Rasa
1	Perceived ease of use	3.35	3.091
2	Perceived usefulness	3.288	3.055
3	Curiosity	3.211	2.955
4	Control	3.125	2.9
5	Joy	3.155	2.911
6	Behavioral intention to use	3.284	3.032
7	Immersion	3.257	3.007
8	All Component	3.239	2.993

Table 9. t-Test: Two-sample assuming equal variances

t-Test: Two-Sample Assuming Equal Variances		
	Variable 1	Variable 2
Mean	3.238571429	2.993
Variance	0.005415102	0.004568857
Observations	8	8
Pooled Variance	0.00499198	
Hypothesized Mean Difference	0	
df	14	
t Stat	6.95138643	
P(T<=t) one-tail	3.37433E-06	
t Critical one-tail	1.761310136	
P(T<=t) two-tail	6.74867E-06	
t Critical two-tail	2.144786688	

4. CONCLUSION

The results of the two methods used to test, namely usability and HMSAM on the Botika Platform and Rasa Platform, show that users prefer the Botika Platform from all aspects of testing. These results also provide an overview of the expectations to be achieved in the study. The implementation of Chatbot technology at Qwerty Bank can at least provide new services that make it easier for customers and prospective customers. In addition, by obtaining a Chatbot Platform that is suitable for Qwerty Bank, it is hoped that in the future all services at the call center can be implemented so that Qwerty Bank can make savings every month for Call Center fees of Rp 625,000,000.00. In addition, these results can also be used as a reference for banks in Indonesia to implement Chatbot technology in accordance with regulations. In the future, it is hoped that it can assist banking in Indonesia in implementing Chatbot Technology which can be used to provide easy service for its customers. In addition, in the future it is still possible for the Chatbot Platform to be implemented for other features or even replace the customer service and call center as a whole, thereby reducing operational costs. In addition, it can also be used as a platform to support the concept of digital banking and even branchless banking.

REFERENCES

[1] L. Cui, S. Huang, F. Wei, C. Tan, C. Duan, and M. Zhou, “SuperAgent: a customer service chatbot for e-commerce websites,” in *Proceedings of ACL 2017, System Demonstrations*, 2017, pp. 97-102, doi: 10.18653/v1/p17-4017.

[2] F. Aloul, S. Zahidi and W. El-Hajj, "Two factor authentication using mobile phones," *2009 IEEE/ACS International Conference on Computer Systems and Applications*, 2009, pp. 641-644, doi: 10.1109/AICCSA.2009.5069395.

[3] R. Luckin, W. Holmes, M. Griffiths, and L. B. Forcier, *Intelligence unleashed: An argument for AI in Education*, London: Pearson Education, 2016.

[4] E. S. Khorasani, "Artificial intelligence: structures and strategies for complex problem solving," *Scalable Computing: Practice and Experience*, vol. 9. no. 3, 2001.

[5] W. Amek, T. Wiegand, and K.-R. Müller, “Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models,” 2019, *arXiv:1708.08296*.

[6] P. Glauner, J. A. Meira, and R. State, "Machine learning for data-driven smart grid applications," in *IEEE Conference on Innovative Smart Grid Technologies, Asia (ISGT Asia 2018)*, 2018.

[7] X. Zhu and A. B. Goldberg, “Introduction to semi-supervised learning,” *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 3, no. 1, pp. 1-130, 2009, doi: 10.2200/S00196ED1V01Y200906AIM006.

[8] K. Morik, B.-E. Kietz, W. Emde, and S. Wrobel, “*Knowledge Acquisition and Machine Learning*,” United States: Morgan Kaufmann Publishers Inc., 1993.

[9] T. R. Gruber and P. R. Cohen, "Design for acquisition: principles of knowledge-system design to facilitate knowledge acquisition," *International Journal of Man-Machine Studies*, vol. 26, no. 2, pp. 143-159, 1987, doi: 10.1016/S0020-7373(87)80088-3.

[10] T. C. W. Lin, "Artificial intelligence, finance, and the law," *Fordham Law Review*, vol. 88, pp. 531, 2019.

[11] T. Lalwani, S. Bhalotia, A. Pal, S. Bisen, and V. Rathod, "Implementation of a Chatbot System using AI and NLP." *International Journal of Innovative Research in Computer Science and Technology (IJIRCST)*, vol. 6, no. 3, 2013, doi: 10.21276/ijircst.2018.6.3.2.

[12] S. A. Abdul-Kader and J. C. Woods, "Survey on chatbot design techniques in speech conversation systems," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 6, no. 7, 2015, doi: 10.14569/ijacsa.2015.060712.

[13] J. Huang, M. Zhou, and D. Yang, "Extracting chatbot knowledge from online discussion forums," in *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2007, vol. 7, pp. 423-428.

[14] M. Qiu *et al*., "AliMe chat: A sequence to sequence and rerank based chatbot engine," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017, vol. 2, pp. 498-503, doi: 10.18653/v1/p17-2079.

[15] I. Hwang, H. Jeon, H. R. Oh, D. Lee, M. Kim, and J. Kim "Chatti: a conversational chatbot platform," in *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[16] D. J. Stoner, L. Ford, and M. Ricci, "Simulating military radio communications using speech recognition and chat-bot technology," *The Titan Corporation Orlando*, 2004.

[17] T. Bocklisch, J. Faulkner, N. Pawlowski, and A. Nichol, "Rasa: open source language understanding and dialogue management," 2017, *arXiv:1712.05181*.

[18] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very deep convolutional networks for natural language processing," 2016, *arXiv: 1606.01781*.

[19] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of cnn and rnn for natural language processing," 2017, *arXiv:1702.01923*.

[20] M. F. McTear and Z. Callejas, *Voice application development for Android*, Packt Publishing, 2013.

[21] B. Swartout and Y. Gil, “EXPECT: Explicit Representations for Flexible Acquisition,” in *Proceedings of the Ninth Knowledge Acquisition forKnowledge-Based Systems Workshop (KAW’95)*, 1995.

[22] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," pp. 2-12, 2018.

[23] D. Braun, A. H. Mendez, F. Mathhes, and M. Langen, "Evaluating natural language understanding services for conversational question answering systems," in *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, 2017, pp. 174-185, doi: 10.18653/v1/W17-5522.

[24] D. Quiñones, C. Rusu, and V. Rusu, "A methodology to develop usability/user experience heuristics,” *Computer Standards and Interfaces*, vol. 59, pp. 109-129, 2018, doi: 10.1016/j.csi.2018.03.002.

[25] J. Axup, S. Viller and N. J. Bidwell, "Usability of a mobile, group communication prototype while rendezvousing," *Proceedings of the 2005 International Symposium on Collaborative Technologies and Systems*, 2005, pp. 24-31, doi: 10.1109/ISCST.2005.1553290.

[26] P. B. Lowry, J. E. Gaskin, N. Twyman, B. Hammer, T. L. Roberts, “Taking ‘fun and games’ seriously: proposing the hedonic-motivation system adoption model (HMSAM),” *Journal of the Association for Information Systems*, vol. 14, no. 11, pp. 617-671, 2012, doi: 10.17705/1jais.00347.

[27] M. Lacity, A. Yan, and S. Khan, "Review of 23 years of empirical research on information technology outsourcing decisions and outcomes," *Proceedings of the 50th Hawaii International Conference on System Sciences*, 2017, doi: 10.24251/HICSS.2017.632.

BIOGRAPHIES OF AUTHORS

Bhakti Prabandyo Wicaksono currently holds a Bachelor of Engineering degree obtained from the Muhammadiyah University of Jakarta in 2012. He started his IT career in the Banking Industry in 2013, until starting in 2019 he was trusted to be the AVP IT Department to head the Digital Platforms and Data Analytics. Currently he is a graduate student at Master of Computer Science in Bina Nusantara University. His research is in the area of natural language processing (NLP). He can be contacted at email: bhakti.wicaksono@binus.ac.id.

Amalia Zahra is a lecturer at the Master of Information Technology, Bina Nusantara University, Indonesia. She received her bachelor's degree in computer science from the Faculty of Computer Science, University of Indonesia (UI) in 2008. She does not have a master's degree. Her Ph.D. was obtained from the School of Computer Science and Informatics, University College Dublin (UCD), Ireland in 2014. Her research interests cover various fields in speech technology, such as speech recognition, spoken language identification, speaker verification, speech emotion recognition, and so on. Additionally, she also has interest in natural language processing (NLP), computational linguistics, machine learning, and artificial intelligence. She can be contacted at email: amalia.zahra@binus.edu.

Text similarity algorithms to determine Indian penal code sections for offence report

Ambrish Srivastav, Shaligram Prajapat
Department of Computer Science, International Institute of Professional Studies (IIPS), Devi Ahilya University (DAVV), Indore, India

Article Info

Article history:

Received Mar 25, 2021
Revised Dec 22, 2021
Accepted Dec 29, 2021

Keywords:

Decision support system
Information retrieval system
Law information system
Natural language processing
Text similarity
Vector space model

ABSTRACT

Taking decisions by comparing two text documents is a new innovative idea. Text documents contain details, rules and information related to a domain. The judiciary system is an area where many textual documents are available. In some documents, rules related to the judiciary are mentioned, such as the Indian penal code (IPC) section documents and other documents like first information report (FIR), and Investigation report. contain details of incidents. Our assumption is that the system can help in making the decision by finding the right IPC Section from the result of text similarity between IPC section document and FIR, investigation report. In this research paper, we preface a new research problem to make decisions to suggest appropriate IPC Section for crime related information from user's input by using vector space model and natural language processing techniques.

Corresponding Author:

Ambrish Srivastav
Department of Computer Science, IIPS, DAVV
139, Khandwa Rd, Indrapuri Colony, Indore, Madhya Pradesh (India) 452001
Email: a.srivastav30@gmail.com

1. INTRODUCTION

The decision support system (DSS) is a computerized program used for decision-making activities aimed at growing the business. Presently, due to the progress in the field of computers, all new documents from different areas are being digitalized. Documents related to the judicial system, such as first information reports (FIRs), investigation reports, and judgments are available digitally, in which we can extract any information by implementing a computerized algorithm. In the past decade, some systems were developed to help with decision making by using text similarity algorithms. This system calculates the similarity between two legal documents by using concept based similarity, multi-dimensional similarity [1] and embedding-based methodologies [2]–[4].

Developing DSS to analyze report and finding appropriate Indian penal code (IPC) section according is a new idea. Whenever there is any crime in the society, its information is given to the police and the police are investigate based on that information. The police prepare a comprehensive report (charge sheet) for the court, which mentions sections of the various IPC related to the crime. Knowledge and experience of the sections of the IPC is required to prepare the charge sheet, on the basis of which a correct and appropriate document is prepared for the court. Apart from the police, some other people or organizations can also be users of the system. A lawyer who re-examines the charge sheet and based on his experience prepares the background of the crime and presents it to the offender or victim's side in court. Reading and understanding documents manually such a difficult and time taking task for everyone. If computer program helps in highlighting important information and checking correctness of result according to rules, it will help to understanding document fastly. A common person or organization can also use this system, with which any crime, deception or violation of rights has taken place. The person or organization has to enter the details of the incident with them in the system.

To use the system, the user will have to enter the information of the incident in the form of natural language text and after analyzing the incident, the system will decide the section of the IPC. Here, we propose a DSS for finding IPC sections (as an appropriate answer) for input of the user. The section of the penal code depends on the various situations, circumstances, some other information of the crime and the definition defined in IPC document. Therefore, analysis of IPC documents and inputs will be necessary. A user may also not write exact word of offense according to penal code document in application, report or query as input then our proposed system finds penal code sections as an appropriate answer and related information for the user. Our idea is to calculate similarity between every sentence of user's input and description of every section of IPC document. According to similarity value, system will suggest list of most appropriate IPC sections for user's input.

In earlier days, DSS was developed for decision making for business purposes, but todays, it is evolving for many fields like healthcare, security, medicine, manufacturing, and engineering. In literature, huge work is available for a variety of decision support systems. In recent years there are many various legal/law information systems developed. Quaresma and Rodrigues have proposed a computational linguistic theory (syntactic, semantic analysis and semantic interpretation) based approach to develop a question-answering system for juridical documents in Portuguese language. Query processing by information retrieval and analysis of documents by information extraction are two modules of this question answering systems (QAS). This system contained complete set of decisions from several Portuguese juridical institutions [5]. Tirpude and Alvi have proposed a keyword-based quality assurance (QA) system for legal documents of Indian laws. For this, the author constructs the corpus and knowledge base from legal documents and prepared question dataset with answer type. This system suggested answer of query on the basis of keywords Indexed term dictionary [6]. Kamdi and Agrawal developed question answering system for IPC sections and Indian amendment laws. This QAS select keywords and question type from query and response according answer stored in corpus. Authors define that problem lies on intersection of two domains: Information retrieval (IR) and natural language processing (NLP) [7]. Sangeetha *et al.* have proposed an information retrieval system is designed to retrieve relevant answers about laws. The user query in a system was processed using natural language processing techniques. This system was designed to face dynamic queries from the user end instead of stored question answers [8].

Text processing is an essential part of every natural language based system. Various machine learning approach like decision tree, nearest neighbors, support vector machines, sparse network of windows, naïve bayes and log-linear model (maximum entropy models) experimented for classification of text [8]–[10]. For identifying part-of-speech tagging, name entities and morphological analysis rules-based techniques, Google directory and hidden markov model were developed [11]–[15]. For identifying and removing stop words from text a latent semantic indexing (LSI), SVM-based approach and deterministic finite automata (DFA) were developed [16]–[18]. For solving the issue of statement formation of systematic question Template-based approach proposed. This approach worked on domain-specific Wh-type questions and imperative questions [19].

Calculating text similarity between two different documents is the main task of my research. Various approaches have been proposed by different authors for this work. Mihalcea *et al*. have proposed a corpus-based and knowledge-based measures method of for measuring the semantic similarity of short texts by exploiting the information that can be drawn from the similarity of the component words [20], [21]. Vector space model (VSM) is used for calculating text similarity of small sentences and paragraphs [22]–[25]. Graph-based text similarity (GBTS) algorithm maps Chinese texts into graphs then calculates the similarity of two texts by comparing their graphs [26]. Xue *et al.* presented a method of text similarity computing to the clinical decision support system. Authors improved TF-IDF algorithm and cosine similarity algorithm by combining with eigenvector associated model to determine the case feature weights [27]. Duan and Xu presented short text similarity algorithm for finding similar police incidents. This algorithm was developed from a novel semantic similarity algorithm word mover'd distance (WMD) [28]. Jo proposed the version of k-nearest neighbor (KNN) which considers similarity among attributes for computing the similarity between feature vectors [29]. Noufa Alnajran *et al*. proposed heuristic driven pre-processing methodology for enhancing the performance of similarity measures in the context of twitter tweets [30].

2. PROPOSED ARCHITECTURE OF SYSTEM

Based on rationales in previous sections, Figure 1 presents architecture of DSS for finding the most suitable IPC Section of user's input. In the first layer of the system, user input will be analyzed using NLP techniques and in the second layer a knowledge base for the IPC section document will be developed. System consists of several components including-

– Component for extraction of offence words and crime related information from the user's input query.
– Components for analyzing crime related information and definition of selected IPC sections.
– Relevance matching component for crime: According to the definition of particular IPC sections.
– Get and show most appropriate IPC sections.

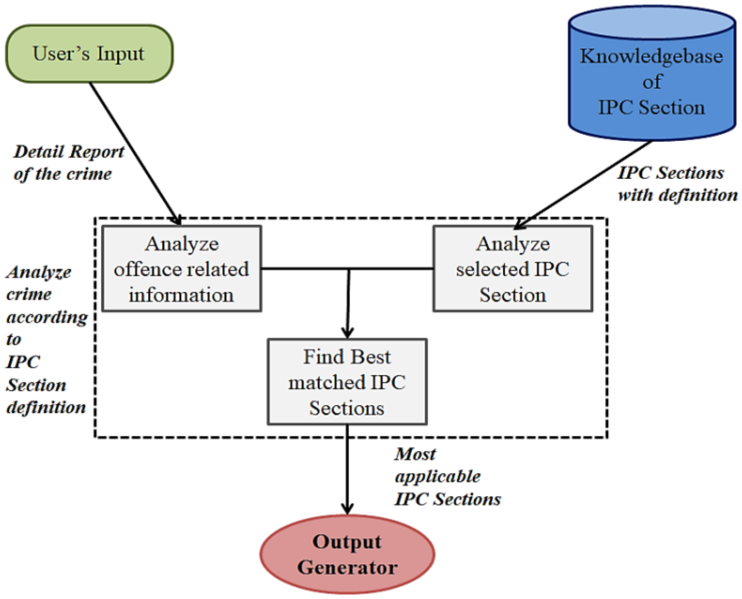


Figure 1. Proposed architecture of system

3. METHOD

IPC document and offence report are two different type of unstructured text. Development of such a system for determines most appropriate IPC Sections for a crime report from unstructured text document of IPC is difficult task. We identify the following steps to achieve our goal.

– Step 1: Developing a corpus for IPC section document. The IPC document distributes 511 sections in 23 chapters. Each chapter describes some kind of crime and conditions. In a corpus of IPC section we include four parts (IPC section no, root, offence and description of section).
– Step 2: Apply method of calculating the text similarity between input text and description of IPC section. Semantic similarity is a measure of conceptual distance between two objects, based on the correspondence of their meanings [31].

The IPC section description text and user input text are two different types of documents and there is very little chance that they are lexical similar. Our objective is to calculate semantic similarity between pair of every sentence of selected IPC section description text with every sentence of user's input. To calculate similarity, follow the following steps:

i) Apply pre-processing in IPC Section description text and user's input text. We used natural language processing toolkit, NLTK for implementing pre-processing. Steps are:
– Tokenization: Tokenization is a procedure of splitting a sentence into list of words.
– Lower casing: Convert all words in common case (most preferable lower case) because in NLP same word in different case treated as a different word.
– Stop words removal: In a text document, there are so many words (like 'is', 'was', 'a', and 'the'.) that do not signify any importance in processing. So, these words must remove from document before processing.
– Stemming/lemmatization: Stemming and lemmatization is a process of transforming a word to its root form. Lemmatization works better then stemming for converting a word to its root form.
– After cleaning text document, we found most important words in IPC section description and user's input for further processing.

ii) Use filtered IPC Section description words as a term. Apply feature engineering for finding feature of user's input text as a vector from term So, feature engineering technique will calculate vector value according to presence of terms or its synonyms word in user's input. There are several techniques that apply to derive relevant features from a text document.

3.1. Vector space model

Vector space model is a matrix representation of list of documents and corpus of words. Every row represents individual document and columns represent words of corpus. Cell store value '0' or '1'. '0' means that word not present in document and '1' indicates word occurred in document. In our problem vector matrix shows occurrence of terms (selected feature of particular IPC section) in a text document (user's input) and according to cells value we can calculate appearance of IPC Section in sentence. In the user's input, there may be many sentences that are not related to the IPC section. If the vector value of all the words in the sentence is '0' then system will ignore that sentence for score calculation. We create vectors for description of each IPC section and every paragraph of user's input and the system will use these vectors for further calculations. There are some tools for converting text document into a vector.

i) CountVectorizer: CountVectorizer is a tool provided by the scikit-learn library in Python. It is used to transform a given text into a vector on the basis of the frequency (count) of each word that occurs in the entire text. Let consider the example for some filtered IPC Section description:

– D0: public nuisance illegal omission cause common injury danger
– D1: unlawfully negligent act likely spread infection disease dangerous life
– D2: malignant act likely spread infection disease dangerous life

Sample result of CountVectorizer shows frequency of words in each document (D0, D1 and D2) in Table 1. If word appears in document then frequency of word becomes '1' otherwise it will be '0'.

ii) TF-IDF: TF-IDF stands for term frequency-inverse document frequency. In this model, we take term frequency and inverse document frequency as parameters to decrease the weight of the terms appearing commonly in all the sentences. Formulas of calculating TF-IDF stepwise are:

– tf(t, d)=count of t in d/number of words in d //term frequency
– df(t)=occurrence of t in documents //document frequency
– idf(t)=log(N/df(t)) //inverse document frequency
– tf-idf(t, d)=tf(t, d)*idf(t)

Sample result of TF-IDF shows frequency of words in each document (D0, D1 and D2) in Table 2. Frequency of each word calculated by its appearance in particular document and all documents.

Table 1. Sample IPC section vector using CountVectorizer

	act	cause	common	danger	public	spread	Unlawfully
0	1	0	0	0	0	1	1
1	1	0	0	0	0	1	0
2	0	1	1	1	1	0	0

Table 2. Sample IPC section vector using TF-IDF

	act	cause	common	danger	public	spread	Unlawfully
0	0.309228	0	0	0	0	0.309228	0.406598
1	0.33847	0	0	0	0	0.33847	0
2	0	0.353553	0.353553	0.353553	0.353553	0	0

– Step 3: Calculate Cosine similarity between vectors of every paragraph of users input with vector of each IPC Section description. Cosine similarity measures the similarity between two vectors of an inner product space as shown in Figure 2. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction. It is often used to measure document similarity in text analysis. Values range between -1 and 1, where -1 is perfectly dissimilar and 1 is perfectly similar.

$$\text{Similarity (A, B)}=\frac{A.B}{||A|| \, X \, ||B||} = \frac{\sum_{i=1}^{n} A_i \, X \, B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \, X \, \sqrt{\sum_{i=1}^{n} B_i^2}}$$

– Step 4: According to this calculation of cosine similarity, system will show list of most appropriate IPC sections that's closely related to users input. Here one document is description of IPC section and another document is paragraph of user's input.

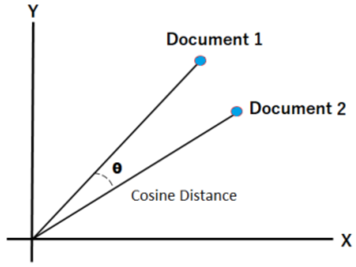


Figure 2. Cosine distance similarity

4. RESULTS AND DISCUSSION

4.1. Development of corpus

There are 511 sections in IPC document thats are devided into 23 chapters. We have selected 4 chapters of the IPC document, which are chapters 14, 15, 16 and 22, to prove the presumed correctness of our proposed work. We developed corpus for sections (around 120) of these chapter as shown in Table 3.

4.2. Select complain for input

We have selected the complaint text as shown in Figure 3 related to these chapters as the input query. These complaints are available in the form of FIR. on the official portal of state police in India. The FIR is divided into paragraphs which contain the offense and its related information.

4.3. Similarity calculation

Count vector and TF-IDF model applied to calculate text similarity between each paragraph of complaint with description of each section and found list of most appropriate '10' IPC sections that's most related to complain as shown in Table 4. As a result both models produce some list of IPC sections. This list and its sequence are different in result of both model but most of sections are common related to complain. Based on the output of these models, the system can act as decision support for the user.

Table 3. Corpus for IPC section document

Section	Root	Offence	Description
268	nuisance	Public nuisance	Public nuisance, illegal omission which causes any common injury, danger
269	negligently	Negligent act	Unlawfully, Negligent act likely to spread infection of disease dangerous to life
270	malignant	Malignant act	Malignant act likely to spread infection of disease dangerous to lifedangerous to life

> Jayendra Saraswathi is the head of the Kanchi Mutt, one of a prominent Hindu monastic institutions in the country. An investigative journalist named Dhanasekaran Prakash in the Tamil weekly Nakkeeran alleged the reasons of the murder being the continuous infuriation by Sankararaman against Jayendrar and Kanchi Mutt. ....

Figure 3. Sample complaint text

Table 4. Comparision of count vector and TF-IDF result

Count Vector Result	TF-IDF Result
related_ipcs_index [118 42 48 41 49 51 66 26 123 43] ('IPC', 364, ':', 'Kidnapping or abducting in order to murder') ('IPC', 303, ':', 'Punishment for murder by life-convict') ('IPC', 307, ':', 'Attempt to murder') ('IPC', 302, ':', 'Punishment for murder') ('IPC', 308, ':', 'Attempt to commit culpable homicide') ('IPC', 310, ':', 'Thug') ('IPC', '320F', ':', 'Grievous hurt') ('IPC', 290, ':', 'Punishment for public nuisance in cases not otherwise provided for') ('IPC', '366B', ':', 'Importation of girl from foreign country') ('IPC', 304, ':', 'Punishment for culpable homicide not amounting to murder')	related_ipcs_index [118 42 66 26 48 123 49 41 141] ('IPC', 364, ':', 'Kidnapping or abducting in order to murder') ('IPC', 303, ':', 'Punishment for murder by life-convict') ('IPC', '320F', ':', 'Grievous hurt') ('IPC', 290, ':', 'Punishment for public nuisance in cases not otherwise provided for') ('IPC', 307, ':', 'Attempt to murder') ('IPC', '366B', ':', 'Importation of girl from foreign country') ('IPC', 308, ':', 'Attempt to commit culpable homicide') ('IPC', 302, ':', 'Punishment for murder') ('IPC', '376C', ':', 'Intercourse by superintendent of jail and remand home') ('IPC', 310, ':', 'Thug')

5. CONCLUSION

This research paper starts with an introduction of a problem in judicial system and finds solution by using decision support system (DSS). DSS aims to help make the best decision based on existing information. Over the past few decades, a number of information retrieval (IR) system and question answering systems (QAS) have been developed to find result and answers in a limited specific area. IR system and QAS takes single line question and apply NLP techniques to extract keyword and search result. Here we propose the architecture of DSS for crime incident documents which suggest the list of most applicable IPC section by comparing the user input document and IPC section document by vector space model. Our proposed system enhances the working of typical question answering system and help to take decision on the basis of result. In the future, some other text similarity algorithms such as word2vec, doc2vec, and BERT (sentence transform). will use to check the acureacy of the system.

ACKNOWLEDGEMENT

I want to thank my supervisor Dr. Shaligram Prajapat, Associate Professor in IIPS DAVV, Indore not only for his continued support but for the motivation and fruitful advises in accomplishing this task.

REFERENCES

[1] R. S. Wagh and D. Anand, "Legal document similarity: a multi-criteria decision-making perspective," *PeerJ Computer Science*, vol. 6, Art. no. e262, Mar. 2020, doi: 10.7717/peerj-cs.262.
[2] A. Mandal, R. Chaki, S. Saha, K. Ghosh, A. Pal, and S. Ghosh, "Measuring similarity among legal court case documents," in *Proceedings of the 10th Annual ACM India Compute Conference on ZZZ-Compute '17*, 2017, pp. 1–9, doi: 10.1145/3140107.3140119.
[3] P. Bhattacharya, K. Ghosh, A. Pal, and S. Ghosh, "Methods for computing legal document similarity: a comparative study," *Computer Science*, Apr. 2020.
[4] S. Renjit and S. M. Idicula, "Similarity in legal texts using document level embeddings," *CUSAT NLP@AILA-FIRE2019*, pp. 25–30, 2019.
[5] P. Quaresma and I. P. Rodrigues, "A question answer system for legal information retrieval," in *Proceedings of the 2005 conference on Legal Knowledge and Information Systems: JURIX 2005: The Eighteenth Annual Conference*, 2005, pp. 91–100.
[6] S. C. Tirpude and D. A. S. Alvi, "Closed domain keyword based question answering system for legal documents of IPC sections Indian laws," *International Journal of Innovative Research in Computer and Communication Engineering*, 2015.
[7] R. P. Kamdi and A. J. Agrawal, "Keywords based closed domain question answering system for Indian penal code sections and Indian amendment laws," *International Journal of Intelligent Systems and Applications*, vol. 7, no. 12, pp. 57–67, Nov. 2015, doi: 10.5815/ijisa.2015.12.06.
[8] D. Sangeetha, R. Kavyashri, S. Swetha, and S. Vignesh, "Information retrieval system for laws," in *2016 Eighth International Conference on Advanced Computing (ICoAC)*, Jan. 2017, pp. 212–217, doi: 10.1109/ICoAC.2017.7951772.
[9] D. Zhang and W. S. Lee, "Question classification using support vector machines," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval-SIGIR '03*, Aug. 2003, p. 26, doi: 10.1145/860435.860443.
[10] P. Blunsom, K. Kocik, and J. R. Curran, "Question classification with log-linear models," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval-SIGIR '06*, 2006, p. 615, doi: 10.1145/1148170.1148282.
[11] J. Liu and L. Birnbaum, "Measuring semantic similarity between named entities by searching the web directory."
[12] R. Ageishi and T. Miura, "Named entity recognition based on a Hidden Markov Model in part-of-speech tagging," in *2008 First International Conference on the Applications of Digital Information and Web Technologies (ICADIWT)*, Aug. 2008, pp. 397–402, doi: 10.1109/ICADIWT.2008.4664380.
[13] Zhang Youzhi, "Research and implementation of part-of-speech tagging based on Hidden Markov Model," in *2009 Asia-Pacific Conference on Computational Intelligence and Industrial Applications (PACIIA)*, Nov. 2009, pp. 26–29, doi: 10.1109/PACIIA.2009.5406648.
[14] R. Cretulescu, A. David, D. Morariu, and L. Vintan, "Part of speech tagging with Naïve Bayes methods," in *2014 18th International Conference on System Theory, Control and Computing (ICSTCC)*, Oct. 2014, pp. 446–451, doi: 10.1109/ICSTCC.2014.6982457.
[15] S. P. Singh, A. Kumar, and H. Darbari, "Deep neural based name entity recognizer and classifier for English language," in *2017 International Conference on Circuits, Controls, and Communications (CCUBE)*, Dec. 2017, pp. 242–246, doi: 10.1109/CCUBE.2017.8394152.
[16] A. N. K. Zaman, P. Matsakis, and C. Brown, "Evaluation of stop word lists in text retrieval using latent semantic indexing," in *2011 Sixth International Conference on Digital Information Management*, Sep. 2011, pp. 133–136, doi: 10.1109/ICDIM.2011.6093315.
[17] S. Xu, G. Cheng, and F. Kong, "Research on question classification for automatic question answering," in *2016 International Conference on Asian Language Processing (IALP)*, Nov. 2016, pp. 218–221, doi: 10.1109/IALP.2016.7875972.
[18] S. Behera, "Implementation of a finite state automaton to recognize and remove stop words in english text on its retrieval," in *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, May 2018, pp. 476–480, doi: 10.1109/ICOEI.2018.8553828.
[19] K. Pawar and U. Shrawankar, "Question systematization using templates," *3rd International Conference on Computing for Sustainable Global Development,* 2016.
[20] R. Mihalcea and C. C. C. Strapparava, "Corpus-based and knowledge-based measures of text semantic similarity," in *{AAAI}'06: {Proceedings} of the 21st national conference on {Artificial} intelligence*, Jul. 2006, vol. 1, pp. 775–780.
[21] W. H.Gomaa and A. A. Fahmy, "A survey of text similarity approaches," *International Journal of Computer Applications*, vol. 68, no. 13, pp. 13–18, Apr. 2013, doi: 10.5120/11638-7118.

[22] H. Dong, J. Wu, X. Zhao, and Y. Li, "Study on the calculation of text similarity based on key-sentence," in *2010 International Conference on E-Business and E-Government*, May 2010, pp. 1952–1955, doi: 10.1109/ICEE.2010.493.

[23] W. Yih, K. Toutanova, J. C. Platt, and C. Meek, "Learning discriminative projections for text similarity measures," in *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, 2011, pp. 247–256.

[24] P. Shrestha, "Corpus-based methods for short text similarity," in *TALN 2011*, 2011, pp. 1–6.

[25] G. Liu and H. Wang, "A recursive descent evaluation algorithm on policy context similarity," in *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*, May 2018, pp. 21–25, doi: 10.1109/ICAIBD.2018.8396160.

[26] Z. Liu and X. Chen, "Mapping texts into graphs: An improved text similarity algorithm," in *Proceedings of 2012 2nd International Conference on Computer Science and Network Technology*, Dec. 2012, pp. 1357–1361, doi: 10.1109/ICCSNT.2012.6526173.

[27] T. Xue, Y. Yuan, Q. Fu, H. Gu, S. Zhang, and C. Wang, "The application of text similarity computing in the clinical decision support system," Nov. 2014, doi: 10.1109/ccis.2014.7175759.

[28] L. Duan and T. Xu, "A short text similarity algorithm for finding similar police 110 incidents," in *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*, Nov. 2016, pp. 260–264, doi: 10.1109/CCBD.2016.058.

[29] T. Jo, "Using k-nearest neighbors for text segmentation with feature similarity," in *2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE)*, Jan. 2017, pp. 1–5, doi: 10.1109/ICCCCEE.2017.7866706.

[30] N. Alnajran, K. Crockett, D. McLean, and A. Latham, "A heuristic based pre-processing methodology for short text similarity measures in microblogs," in *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, Jun. 2018, pp. 1627–1633, doi: 10.1109/HPCC/SmartCity/DSS.2018.00265.

[31] D. Lin, "An information-theoretic definition of similarity," in *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, 1998, pp. 296–304.

BIOGRAPHIES OF AUTHORS

Ambrish Srivastav is a research scholar at, Devi Ahilya University (DAVV), Indore and having teaching experience approx 10 years in field of Computer Science and Engineering. He graduated in 2009 from I.E.T.E, New Delhi and received his Master's degree in 2011 from I.E.T. DAVV. His research interests are Artificial Intelligence, Natural Language Processing and Machine Learning. He can be contacted at email: a.srivastav30@gmail.com.

Dr. Shaligram Prajapat has been working in academics as an educationist, teacher, researcher and learner since past 2 decades. He has executed many academic and research projects as a part of Devi Ahilya university, India. In research career, with Ph.D. from Maulana Azad National Institute of Technology (M.A.N.I.T.) in Computer Applications from Bhopal India and Master of Philosophy (Computer Science) from Devi Ahilya University Indore, he has more many research publications in International journals listed in web of science and Scopus. He can be contacted at email: shaligram.prajapat@gmail.com.

IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 11, No. 1, March 2022, pp. 41~49
ISSN: 2252-8938, DOI: 10.11591/ijai.v11.i1.pp41-49

Solving a traveling salesman problem using meta-heuristics

Anahita Sabagh Nejad, Gabor Fazekas
Department of Informatics, University of Debrecen, Debrecen, Hungary

Article Info

Article history:

Received Mar 25, 2021
Revised Dec 16, 2021
Accepted Dec 30, 2021

Keywords:

Clustering method
Traveling salesman problem
meta-heuristics
Whale optimization algorithm

ABSTRACT

In this article, we have introduced an advanced new method of solving a traveling salesman problem (TSP) with the whale optimization algorithm (WOA), and K-means which is a partitioning-based algorithm used in clustering. The whale optimization algorithm first was introduced in 2016 and later used to solve a TSP problem. In the TSP problem, finding the best path, which is the path with the lowest value in the fitness function, has always been difficult and time-consuming. In our algorithm, we want to find the best tour by combining it with K-means which is a clustering method. In other words, we want to divide our problem into smaller parts called clusters, and then we join the clusters based on their distances. To do this, the WOA algorithm, TSP, and K-means must be combined. Separately, the WOA-TSP algorithm which is an unclustered algorithm is also implemented to be compared with the proposed algorithm. The results are shown through some figures and tables, which prove the effectiveness of this new method.

Corresponding Author:

Anahita Sabagh Nejad
Department of Informatics, University of Debrecen
4028 Debrecen, Kassai Street 26, Hungary
Email1: Anahita.sabagh@inf.unideb.hu

1. INTRODUCTION

Solving the traveling salesman problem (TSP) has always been one of the topics of interest to researchers for example solving the vehicle routing problem that was first introduced by Dantzig *et al.* [1], [2]. The problem of determining the travel route is similar to finding the solution from the TSP problem [3]. Since TSP is an NP-hard problem and it takes a protracted time to search out a tour among the cities, the complexity order of this problem becomes exponential which does not have a suitable execution time [4], but an intelligent method requires less computation time and more accuracy [5]. Metaheuristic algorithms are powerful methods for solving many tough optimization problems [6]. In the past three decades, meta-heuristics pose a potential impact on tasks of managing operational problems (e.g. assignment and scheduling) [7].

Swarm intelligence is an optimization technique that looks for the domain by developing a population of individuals. These individuals represent solutions for our problems [8]. The individuals move to the better solution areas. Each iteration forces the individuals to cooperate till reaching the best solution [8]–[10]. The swarm-based algorithms have been proven to be effective in solving nonlinear optimization problems in a large space search domain [11]–[13]. Over the last decades, nature has been a source of inspiration for several meta-heuristics, which have been introduced to solve optimization problems. These meta-heuristics have been tested to solve discrete problems [14]. Some of the nature- inspired algorithms are: particle swarm optimization (PSO) [15], [16], ant colony optimization (ACO) [17]–[19], artificial bee colony (ABC) [20], cuckoo search (CS) algorithm [21], [22], krill herd (KH) [23], and spotted hyena optimizer (SHO) [24].

Exploration and extraction are two basic factors that control a metaheuristic algorithm. Exploration alludes to a metaheuristic algorithm's capacity to search in diverse environment ranges to find the ideal solutions [25]–[27]. On the other hand, exploration is the ability to centralize search inside the optimal range to extract the optimal solution. A metaheuristic algorithm balances these two clashing objectives so that in any metaheuristic calculation or the progressed version, performance is moved forward by controlling the aforementioned parameters [27].

In [28], a new clusterized technique was introduced based on K-means to solve TSP by using the firefly algorithm which we extended that by using the WOA as a new studying. In TSP, with the view to minimize the cost function and hence improve the efficiency, the WOA is combined with the K-means. The same technique is applied to the unclustered algorithm, but based on the characteristics of the TSP problem [29] and the mechanism of the WOA, The maximum scale is determined to have 1323 cities, because the unclustered method is not as good as the clustered method, and we wanted to use the same datasets for both. The figures prove this in the Subsection 4.2. Section 2 provides some detailed information about the TSP, K-means, WOA algorithm, and mathematical models. In section 3, the proposed method is explained, while the results and discussions of the proposed method have been provided in section 4. Finally, the conclusion has been discussed in section 5.

2. THE COMPREHENSIVE THEORETICAL BASIS

In the following, we explain some theories in the subsection parts, which are numbered 2.1 to 2.4 for the TSP problem, K-means, WOA, and mathematical models respectively. The mathematical models have 3 types that are explained with a peseodocode. These types are encircling Prey, bubble-Net attacking method, exploration phase or search for the prey.

2.1. The travelling salesman problem (TSP)

In the theory, the problem is explained on a graph G=(V, A), where we have n vertices $V = \{\upsilon 1,\ \upsilon 2, \ldots, \upsilon n\}$, and a distance matrix $C = (c_{ij})$. We have described **A** as a set of arcs [30] in the (1). In TSP problem, our objective function is the tour with minimum distance. If we want to solve that we need $\frac{(n-1)!}{2}$ [31] comparisons, which makes it impossible to be solved theoretically. We want to improve the time and the complexity of this problem by using the WOA algorithm discovered by Mirjalili, and a data mining technique called K-means. Since TSP belongs to the vehicle routing problem (VRP) and has a minimum cost objective function [32], we apply heuristic algorithms to find the best solutions.

$$A = \{(\upsilon_i,\ \upsilon_j) \big|\ \upsilon_i,\ \upsilon_j \in\ V,\ i \neq j\} \tag{1}$$

2.2. K-means algorithm

Clustering is a method that partitions our data into some groups, so-called clusters [33]–[38]. The steps of data mining start with K clusters, for K centroids. This approach which is iterative starts with a random selection of K objects for K clusters that is the first assignment. As it is explained in the algorithm below, the algorithm should compute the average of the objects for the new assignments of objects to the nearest cluster.

Algorithm: K-means,

```
Inputs:
K: cluster numbers or the initial centroids,
D: a set of n objects,
Output: K clusters,
Method:
(1) arbitrarily choose K objects from D,
(2) repeat,
(3) according to the average of the objects within the cluster, (re)assign the objects to
the closest one
(4) update by calculating the average of the objects for each cluster, and new assignments
(5) until no change happens in the clusters.
```

2.3. The whale optimization algorithm

This algorithm is a type of swarm intelligence algorithm which is inspired by the characteristics of whales [39]. The hunters look for a group of prey to encircle and gradually tighten the ring until they catch that prey [40]. Whales have a special hunting method that is called the bubble-net feeding method which is done by creating a distinctive bubble along a circle [41].

2.4. Mathematical models

Three mathematical models exist that based on them we explain the equations. These models are encircling prey, bubble-net attacking method (spiral bubble net feeding maneuver), and search for the prey. The equations are addressed in the pseudocode of the WOA algorithm. In each model, the D and X parameters obtain new values based on their new positions and distances.

2.4.1. Encircling prey

Whales don't know the best position at the first step, so the target prey becomes the current best candidate solution, but in the next iterations after finding the best agent, the other search agents will try to change their positions to the best candidate solutions. The best search agent is the closest one. The other whales change their positions toward to the best answer. Based on Mirjalili and all, we have the (2) to (5):

$$\overrightarrow{D'} = |\vec{C}.\overrightarrow{X*}(t) - \vec{X}(t)| \tag{2}$$

$$\vec{X}(t+1) = \overrightarrow{X*}(t) - \vec{A}.\vec{D} \tag{3}$$

$$\vec{A} = 2\vec{a}.\vec{r} - \vec{a} \tag{4}$$

$$\vec{C} = 2.\vec{r} \tag{5}$$

D is the distance from (2), but *A* and *C* are coefficients, A=[-a, a] where a is a decreasing number from 2 to 0, and X represents the whale's position, which we use as the agent, r is a number that is selected randomly in [0, 1], *X(t+1)* is the next position for the next whale, which can be a better answer, and t indicates to the iterations. *X*(t)* is the best solution among the whales that we have searched so far, and my change later.

2.4.2. Bubble-net attacking method

We have these methods; i) shrinking encircling mechanism, and ii) spiral updating position (exploration). In another word, The WOA algorithm adopted these two approaches to update the positions of the whales. The first method is realized by setting a by (4), except for the second approach, that's spiral updating position, we've got the following equations, where D is the distance, b indicates a constant for logarithmic shape, and l stands for a value in this interval [-1,1].

$$\vec{X}(t+1) = \overrightarrow{D'}.e^{bl}.cos(2\pi l) + \overrightarrow{X*}(t) \tag{6}$$

$$\overrightarrow{D'} = \overrightarrow{X*}(t) - \vec{X}(t) \tag{7}$$

Selection between these two approaches is based on the p-value that is a random number in [0, 1]. This is for updating or changing the position. If $p < 0.5$ we apply the (3) for X(t+1), otherwise if $p \geq 0.5$, we apply (6), as it is summarized (8):

$$\vec{X}(t+1) = \begin{cases} \overrightarrow{X*}(t) - \vec{A}.\vec{D} & \text{if } p < 0.5) \\ \overrightarrow{D'}.e^{bl}.cos(2\pi l) + \overrightarrow{X*}(t) & \text{if } p \geq 0.5 \end{cases} \tag{8}$$

2.4.3. Exploration phase

If $|A| \geq 1$, the whales select a random whale. This step is called exploration, and they update their distances and positions by (9) and (10). Based on the new $\vec{X}_{rand}$, which is the position of a random whale, the values of $\vec{D}$, and $\vec{X}(t+1)$ change, as it is addressed in step 12 of the pseudocode. The parameter of a changes in an interval from 2 to zero to give the functionality of searching for the prey. This search is global.

$$\vec{D} = |\vec{C}.\vec{X}_{rand} - \vec{X}| \tag{9}$$

$$\vec{X}(t+1) = \vec{X}_{rand} - \vec{A}.\vec{D} \tag{10}$$

2.4.4. The pseudocode of the whale optimization algorithm (WOA)

The pseudocode which is defined as follows describes the summarization of the above-mentioned mathematical models for WOA. After defining the population size, as each whale is an agent, we should find

its fitness. In this problem, our fitness is minimized. Before meeting the termination criterion in the proposed method, we update the values, and according to the equations, and conditions which we described earlier, we find the optimal position of the best whale with minimum fitness.
1: Initialize the population X_i (i=1, 2, … n)
2: Calculate the fitness of each search agent
3: X*=the best search agent
4: While (t < maximum number of iterations)
5: for each search agent
6: Update a, A, C, l, and p
7: If1 ($p < 0.5$)
8: if2 ($|A| < 1$)
9: Update the position of the current search agent by the (3)
10: else if2 ($|A| \geq 1$)
11: Select a random search agent (X_{rand})
12: Update the position for the current agent by the (10)
13: else if1($p \geq 0.5$)
14: Update the position of the current search by the (6)
15: Check if any search agent exists beyond the search space and amend it
16: Calculate the fitness of each search agent
17: Update X*

3. THE PROPOSED METHOD AND ALGORITHM

In this work, we are proposing a method that combines K-means with WOA to solve TSP. This new TSP solver finds the number of clusters based on (11), where N is the number of cities [37]. First, we apply K-means to divide our data into K clusters, then we apply the WOA algorithm to find a tour in each cluster which is minimum, and in the final step, we connect our clusters to find the best solution that is the optimal path.

$$K = \sqrt{\frac{N}{2}} \quad (11)$$

We apply the next [28] procedure:
i) We calculate the distances between the centroids of each cluster to find the clusters with minimum distances (the (12) for finding the distance).
ii) We combine the selected clusters in one bigger cluster [37]. Then we connect the tours of these two newfound clusters.
iii) We repeat step 1 until the generation of the minimum tour.

$$d_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \quad (12)$$

In this method, each whale is an agent and can be a solution for TSP. The algorithm updates to find the current best solutions. In this algorithm, we consider 4 candidate nodes in step 6. These nodes are around the centroids of that cluster, and just the closer one will be joined to another cluster so that we always select the nearest city. The steps of our algorithm are:
- Step 1: Initialize the number of population as shown in Table 1
- Step 2: Specify K based on the (11)
- Step 3: Applying K-means algorithm
- Step 4: Applying whale optimization algorithm for i=1: K
- Step 5: Find the position of the cities
- Step 6: Sorting by indexing
- Step 7: Find the nodes (cities) in each cluster that are closer to the centroid of that clusters
- Step 8: Join the closest cities to another cluster
- Step 9: Stopping criterion till no cluster remains unjoined

4. RESULTS AND DISCUSSIONS

The platform used for the implementation of this method is matrix laboratory (MATLAB), using Intel CORE i5, and 6 GB RAM. On the last page, our graphical illustrations and tables show the complexity of our algorithm before, and after applying K-means. Our datasets are Eil51, Linhp 318, and rl1323. All data are available in the standard TSP library [42]. The vector a is assigned to have the maximum value of 2 for all of the iterations.

4.1. Tables results before and after applying the clustering method

In these tables, the average of our best tour (fitness) is considered as the benchmark. Table 1 shows some parameters. The initial population and iterations have the same value for both algorithms and are chosen based on some experimental results during the execution of the program, but they can have different values as well. Table 2 represents the results of the fitness function before applying K-means clustering method. We have three datasets, and for each, we have calculated some statistics consisting of the min, max, average, and standard deviation of the fitness function. The fitness average for the first approach has the values of 1176.42, 393928.2, and 8262213. Table 3 indicates the fitness function of the clustered method, which is done by using the K-means approach. The results of the comparison between these two tables prove that the fitness average for the first approach has improved to 489.8785, 82908.2, and 1.19E+06 for the second approach for the same datasets. The statistical values of these two tables show how the fitness function improves when our algorithm combines with K-means, especially for the third dataset. Table 4 shows the execution time of the unclustered approach, which means solving a TSP problem using only the whales and without clustering. The time average has values of 3.765785 (s), 16.63365 (s), and 63.50511 (s), which shows high values especially for the third dataset with 1323 nodes (cities). As in Table 4, even the best value (min) for the third dataset takes 51.669 (s) to run. Table 5 shows the execution time for the clustered approach. Based on this table, time has improved since the time average is reduced from 3.765785 (s), 16.63365 (s), and 63.50511 (s) to 1.2205855(s), 5.707375 (s), and 24.81385 (s) for the clustered method. The minimum value for the third dataset takes 20.345 (s) to run. It confirms that our algorithm is improved more than 50%.

Table 1. The parameter settings

Initial Population	Iterations	Number of the Cities
100	20	51
100	20	318
100	20	1323

Table 2. The fitness value for the unclustered whale optimization algorithm

Dataset	Min	Max	Average	Stdev.s
Eil51	1100.269	1262.614	1176.421	48.53605
Linhp318	376614.9	409489.4	393928.2	8468.606
Rl1323	8174234.322	8740947	8262213	49003.98

Table 3. The fitness value for the clustered whale optimization algorithm

Dataset	Min	Max	Average	Stdev.s
Eil51	454.9	526.76	489.8785	21.2760405
Linhp318	77273	88040	82908.2	3180.155
Rl1323	1.14E+06	1.23E+06	1.19E+06	2.78E+04

Table 4. The execution time for the unclustered whale optimization algorithm

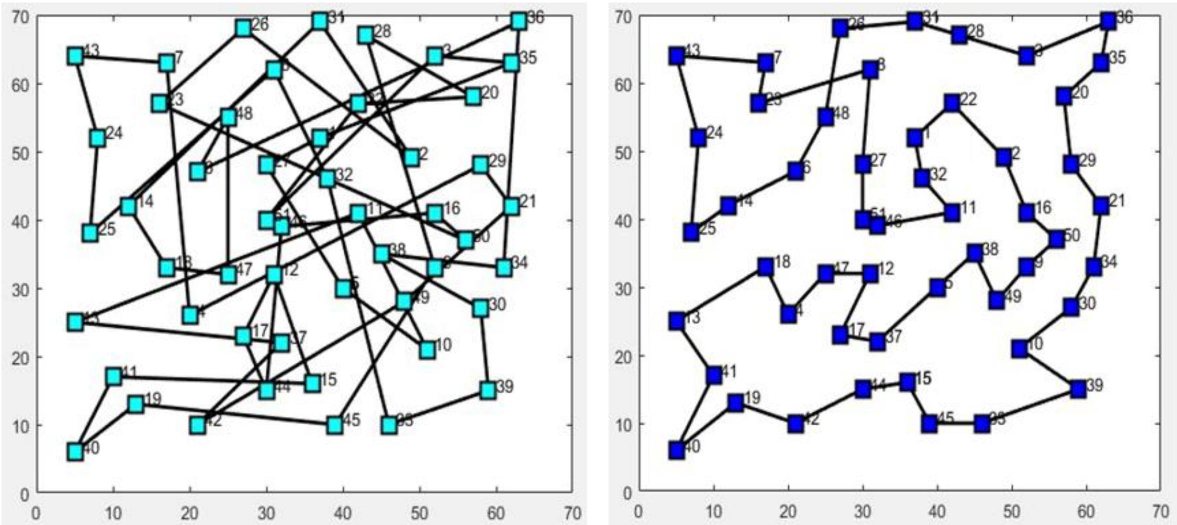
Dataset	Min	Max	Average	Stdev.s
Eil51	3.1421	4.7562	3.765785	o.396958
Linhp318	13.642	20.79	16.63365	1.77977
Rl1323	51.669	79.21	63.50511	6.286887

Table 5. The execution time for the clustered whale optimization algorithm

Dataset	Min	Max	Average	Stdev.s
Eil51	1.0021	1.4997	1.2205855	0.1708681
Linhp318	4.6754	7.953	5.707375	0.979334
Rl1323	20.345	55.258	24.81385	7.647444

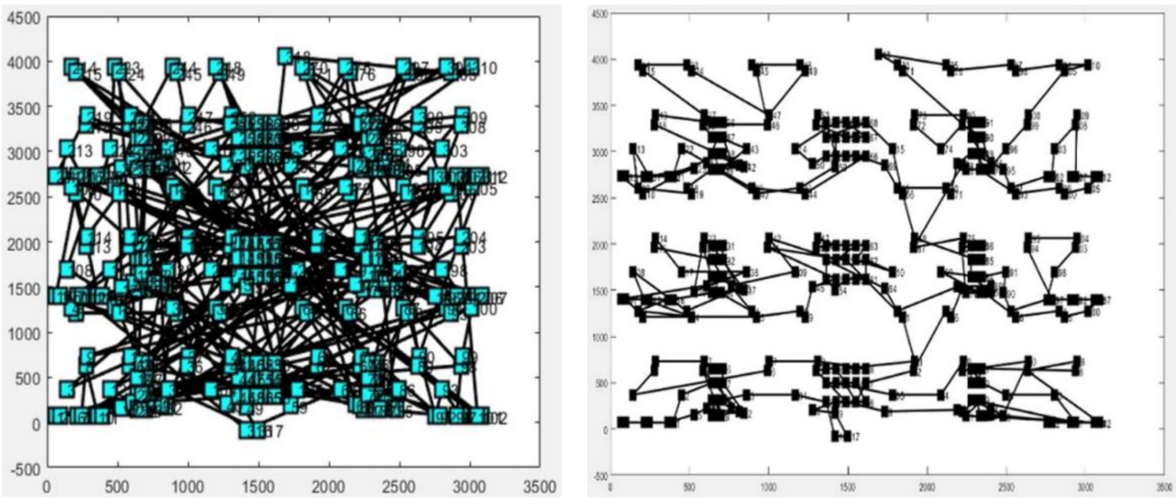
4.2. Figures

In this section, the figures show how this designed method is used to find a more efficient route so that all nodes are met once, and the salesman returns to the starting point. As it is obvious, the figures become more complex as the number of nodes becomes more, so the illustration of all the connections becomes hard as well, therefore the first illustration which is Figure 1 for the Eil51 dataset is a clearer indication of the calculations that have been done. Figure 1(a) shows a graph, which demonstrates the routes of the salesman from the time when our algorithm starts. Figure 1(b) demonstrates the salesman routing but with the help of clustering, so this figure finds the connections for the smallest distances, which are more optimized. In Figure 2 the TSP is solved for Linhp318 such that in (a) unclustered method and (b) clustered approach are presented. Figure 2(a) is more complicated than Figure 2(b). The number of the dataset is 318, but in Figure 2(b), the problem is solved with less complexity, and the reason is using the K-means clustering approach. This approach is helped to find better connections.



(a) (b)

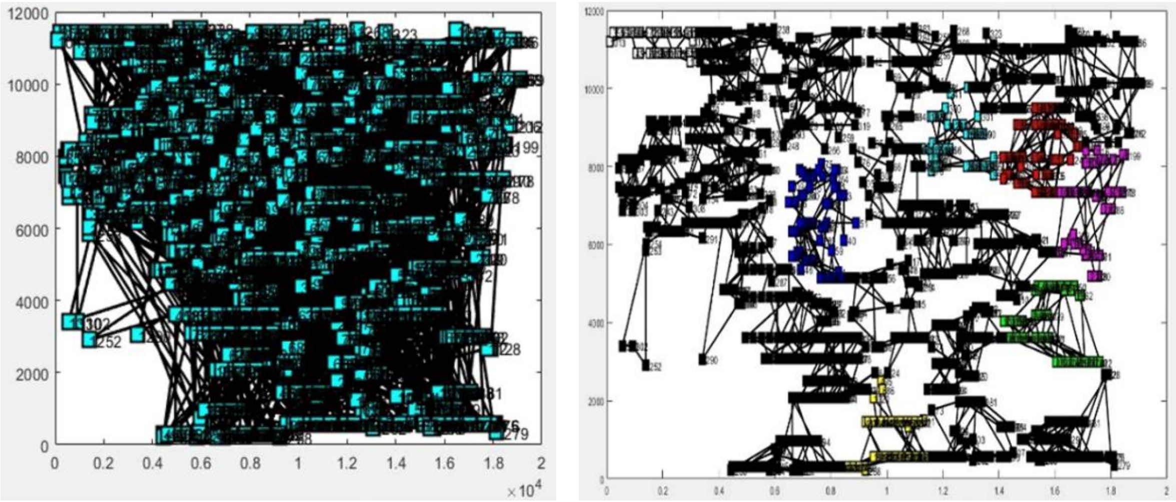
Figure 1. Comparison between different approaches for the Eil51 dataset (a) unclustered approach and (b) clustered approach



(a) (b)

Figure 2. Comparison between different approaches for the Linhp318 dataset (a) unclustered approach and (b) clustered approach

Figure 3 compares the results in the algorithm performance for the rl1323 dataset. In (a) unclustered method and (b) clustered approach are applied. Figure 3(a) shows more complexity than the previous figures. The reason is the number of the dataset with 1323 nodes, which means the salesman should travel between all these nodes and return to the first node. Solving such a problem with this amount of data practically is impossible but with the help of whales, it became solved with high complexity in Figure 3(a). Figure 3(b) is solved this problem with less complexity because of the clustering approach.



(a) (b)

Figure 3. Comparison between different approaches for the Rl1323 dataset (a) unclusterd approach and (b) clustered approach

5. CONCLUSION AND FUTURE WORKS

This article introduced two approaches to solve a TSP: an unclustered method, and a combinatorial approach based on WOA, and K-means. The WOA algorithm, and K-means are combined to solve a TSP problem. This approach divides the problem into some clusters and applies WOA for these small clusters. In the end, this combined algorithm joins the nearest clusters. The results generate the optimal solution concerning the iterative best cost. This new clusterized heuristic has proven to dominate the first approach of solving the same problem based on mentioned tables, and figures. Future research will be for solving more high-scale optimization problems. Furthermore, this article is a prime studying for future research on clustering since the WOA algorithm can be hybridized with other clustering algorithms to find more efficient solutions than the proposed method.

ACKNOWLEDGEMENTS

I would like to thank my university for supporting this work, and my supervisor Dr. Gabor Fazekas for the encouragement, and guidance he has provided throughout my research.

REFERENCES

[1] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, no. 1, pp. 80–91, Oct. 1959, doi: 10.1287/mnsc.6.1.80.

[2] N. Alsumairat and M. Alrefaei, "Solving hybrid-vehicle routing problem using modified simulated annealing," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 6, pp. 4922–4931, Dec. 2021, doi: 10.11591/ijece.v11i6.pp4922-4931.

[3] Z. K. A. Baizal, K. M. Lhaksmana, A. A. Rahmawati, M. Kirom, and Z. Mubarok, "Travel route scheduling based on user's preferences using simulated annealing," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 2, pp. 1275–1287, Apr. 2019, doi: 10.11591/ijece.v9i2.pp1275-1287.

[4] E. Damghanijazi and A. Mazidi, "Meta-heuristic approaches for solving travelling salesman problem," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 5, pp. 18–23, 2017.

[5] K. Loubna, B. Bachir, and Z. Izeddine, "Ant colony optimization for optimal low-pass state variable filter sizing," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 1, pp. 227–235, Feb. 2018, doi: 10.11591/ijece.v8i1.pp227-235.

[6] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO*

2010), Springer Berlin Heidelberg, 2010, pp. 65–74.
[7] Y. K. Qawqzeh *et al.*, "Applying the big bang-big crunch metaheuristic to large-sized operational problems," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 3, pp. 2484–2502, Jun. 2020, doi: 10.11591/ijece.v10i3.pp2484-2502.
[8] H. M. Hasanien, "Performance improvement of photovoltaic power systems using an optimal control strategy based on whale optimization algorithm," *Electric Power Systems Research*, vol. 157, pp. 168–176, Apr. 2018, doi: 10.1016/j.epsr.2017.12.019.
[9] C. Dhaenens and L. Jourdan, *Metaheuristics for big data*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2016.
[10] A. S. Jaradat and S. B. Hamad, "Community structure detection using firefly algorithm," *International Journal of Applied Metaheuristic Computing*, vol. 9, no. 4, pp. 52–70, Oct. 2018, doi: 10.4018/IJAMC.2018100103.
[11] Z. Beheshti and S. M. Shamsuddin, "A review of population-based meta-heuristic algorithm," *International Journal of Advances in Soft Computing and its Applications*, vol. 5, no. 1, pp. 1–35, 2013.
[12] E. Hazir, E. S. Erdinler, and K. H. Koc, "Optimization of CNC cutting parameters using design of experiment (DOE) and desirability function," *Journal of Forestry Research*, vol. 29, no. 5, pp. 1423–1434, Sep. 2018, doi: 10.1007/s11676-017-0555-8.
[13] M. Song and D. Chen, "An improved knowledge-informed NSGA-II for multi-objective land allocation (MOLA)," *Geo-spatial Information Science*, vol. 21, no. 4, pp. 273–287, Oct. 2018, doi: 10.1080/10095020.2018.1489576.
[14] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, May 2016, doi: 10.1016/j.advengsoft.2016.01.008.
[15] T. Zeugmann *et al.*, "Particle swarm optimization," in *Encyclopedia of Machine Learning*, Boston, MA: Springer US, 2011, pp. 760–766.
[16] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, pp. 1945–1950, doi: 10.1109/CEC.1999.785511.
[17] M. Dorigo and M. Birattari, "Ant colony optimization," in *Encyclopedia of Machine Learning and Data Mining*, Boston, MA: Springer US, 2017, pp. 56–59.
[18] M. Dorigo and G. Di Caro, "Ant colony optimization: a new meta-heuristic," in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, 1999, pp. 1470–1477, doi: 10.1109/CEC.1999.782657.
[19] Sourabh Joshi and Sarabjit Kaur, "Ant colony optimization meta-heuristic for solving real travelling salesman problem," in *Emerging Research in Computing, Information, Communication and Applications*, Singapore: Springer Singapore, 2016, pp. 55–63.
[20] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, Oct. 2007, doi: 10.1007/s10898-007-9149-x.
[21] H. Chiroma *et al.*, "Bio-inspired computation: Recent development on the modifications of the cuckoo search algorithm," *Applied Soft Computing*, vol. 61, pp. 149–173, Dec. 2017, doi: 10.1016/j.asoc.2017.07.053.
[22] X.-S. Yang and S. Deb, "Cuckoo search: recent advances and applications," *Neural Computing and Applications*, vol. 24, no. 1, pp. 169–174, Jan. 2014, doi: 10.1007/s00521-013-1367-1.
[23] A. H. Gandomi and A. H. Alavi, "Krill herd: A new bio-inspired optimization algorithm," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 12, pp. 4831–4845, Dec. 2012, doi: 10.1016/j.cnsns.2012.05.010.
[24] S. Ghafori and F. S. Gharehchopogh, "Advances in spotted hyena optimizer: a comprehensive survey," *Archives of Computational Methods in Engineering*, Jul. 2021, doi: 10.1007/s11831-021-09624-4.
[25] N. Rana, M. S. A. Latiff, S. M. Abdulhamid, and H. Chiroma, "Whale optimization algorithm: a systematic review of contemporary applications, modifications and developments," *Neural Computing and Applications*, vol. 32, no. 20, pp. 16245–16277, Oct. 2020, doi: 10.1007/s00521-020-04849-z.
[26] N. Al-Madi, H. Faris, and S. Mirjalili, "Binary multi-verse optimization algorithm for global optimization and discrete problems," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 12, pp. 3445–3465, Dec. 2019, doi: 10.1007/s13042-019-00931-8.
[27] Z. Zhang, C. Huang, K. Dong, and H. Huang, "Birds foraging search: a novel population-based algorithm for global optimization," *Memetic Computing*, vol. 11, no. 3, pp. 221–250, Sep. 2019, doi: 10.1007/s12293-019-00286-1.
[28] A. Jaradat, B. Matalkeh, and W. Diabat, "Solving traveling salesman problem using firefly algorithm and k-means clustering," in *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, Apr. 2019, pp. 586–589, doi: 10.1109/JEEIT.2019.8717463.
[29] S. Lin, "Computer solutions of the traveling salesman problem," *Bell System Technical Journal*, vol. 44, no. 10, pp. 2245–2269, Dec. 1965, doi: 10.1002/j.1538-7305.1965.tb04146.x.
[30] C. Rego, D. Gamboa, F. Glover, and C. Osterman, "Traveling salesman problem heuristics: Leading methods, implementations and latest advances," *European Journal of Operational Research*, vol. 211, no. 3, pp. 427–441, Jun. 2011, doi: 10.1016/j.ejor.2010.09.010.
[31] F. Chebihi, M. essaid Riffi, A. Agharghor, S. Cherif Bourki Semlali, and A. Haily, "Improved chicken swarm optimization algorithm to solve the travelling salesman problem," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 12, no. 3, pp. 1054–1062, Dec. 2018, doi: 10.11591/ijeecs.v12.i3.pp1054-1062.
[32] C. Archetti, D. Feillet, M. Gendreau, and M. Grazia Speranza, "Complexity of the VRP and SDVRP," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 5, pp. 741–750, Aug. 2011, doi: 10.1016/j.trc.2009.12.006.
[33] P. Berkhin, "A survey of clustering data mining techniques," in *Grouping Multidimensional Data*, Berlin/Heidelberg: Springer-Verlag, 2006, pp. 25–71.
[34] R. Xu and D. WunschII, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, May 2005, doi: 10.1109/TNN.2005.845141.
[35] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, Sep. 1999, doi: 10.1145/331499.331504.
[36] P. Kang and S. Cho, "K-means clustering seeds initialization based on centrality, sparsity, and isotropy," in *Intelligent Data Engineering and Automated Learning-IDEAL 2009*, Springer Berlin Heidelberg, 2009, pp. 109–117.
[37] J. Han, M. Kamber, and J. Pei, "Cluster analysis," in *Data Mining*, Elsevier, 2012, pp. 443–495.
[38] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, 1967, pp. 281–297.
[39] J. Zhang, L. Hong, and Q. Liu, "An improved whale optimization algorithm for the traveling salesman problem," *Symmetry*, vol. 13, no. 1, Art. no. 48, Dec. 2020, doi: 10.3390/sym13010048.
[40] R. Oftadeh, M. J. Mahjoob, and M. Shariatpanahi, "A novel meta-heuristic optimization algorithm inspired by group hunting of animals: Hunting search," *Computers and Mathematics with Applications*, vol. 60, no. 7, pp. 2087–2098, Oct. 2010, doi:

10.1016/j.camwa.2010.07.049.
[41] W. A. Watkins and W. E. Schevill, "Aerial observation of feeding behavior in four baleen whales: eubalaena glacialis, balaenoptera borealis, megaptera novaeangliae, and balaenoptera physalus," *Journal of Mammalogy*, vol. 60, no. 1, pp. 155–163, Feb. 1979, doi: 10.2307/1379766.
[42] "MP-TESTDATA-the TSPLIB symmetric traveling salesman problem instances." MP-TESTDATA-The TSPLIB Symmetric Traveling Salesman Problem Instances (accessed Feb. 13, 2021).

BIOGRAPHIES OF AUTHORS

Anahita Sabagh Nejad is a Ph.D. Candidate at the University of Debrecen, Faculty of Informatics, with a background in Computer Science. She is interested in meta-heuristics and has experience in data mining, networking, and the theory of languages and machines as well. She can be contacted at email: Anahita.sabagh@inf.unideb.hu.

Dr. Gabor Fazekas is an Associate Professor at University of Debrecen, Faculty of Informatics, Department of Information Technology and its Theoretical Background. He has several experiences in supervising Ph.D. students. He is retired but still supervising Ph.D. students in different fields like computer systems, databases, combinatorial coding, didactic, and many more. He can be contacted at email: Fazekas.gabor@inf.unideb.hu.

IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 11, No. 1, March 2022, pp. 50~64
ISSN: 2252-8938, DOI: 10.11591/ijai.v11.i1.pp50-64

Comparison of meta-heuristic algorithms for fuzzy modelling of COVID-19 illness' severity classification

Nur Azieta Mohamad Aseri[1], Mohd Arfian Ismail[1], Abdul Sahli Fakharudin[1,2], Ashraf Osman Ibrahim[2,3], Shahreen Kasim[4], Noor Hidayah Zakaria[5], Tole Sutikno[6]

[1]Faculty of Computing, Universiti Malaysia Pahang, Pahang, Malaysia
[2]Faculty of Computer Science and Information Technology, Alzaiem Alazhari University, Khartoum North, Sudan
[3]Arab Open University, Khartoum, Sudan
[4]Faculty of Computing Science and Information Technology, Universiti Tun Hussein Onn, Johor, Malaysia
[5]School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, Johor, Malaysia
[6]Department of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta, Indonesia

Article Info

Article history:

Received Sep 6, 2021
Revised Sep 22, 2021
Accepted Oct 28, 2021

Keywords:

COVID-19
Differential evolution
Fuzzy logic
Genetic algorithm
Meta-heuristic
Particle swarm optimization
TLBO Algorithm

ABSTRACT

The world health organization (WHO) proclaimed the COVID-19, commonly known as the coronavirus disease 2019, was a pandemic in March 2020. When people are in close proximity to one another, the virus spreads mostly through the air. It causes some symptoms in the affected person. COVID-19 symptoms are quite variable, ranging from none to severe sickness. As a result, the fuzzy method is seen favourably as a tool for determining the severity of a person's COVID-19 sickness. However, when applied to a large situation, manually generating a fuzzy parameter is challenging. This could be because of the identification of a large number of fuzzy parameters. A mechanism, such as an automatic procedure, is consequently required to identify the right fuzzy parameters. The meta-heuristic algorithm is regarded as a viable strategy. Five meta-heuristic algorithms were analyzed and utilized in this article to classify the severity of COVID-19 sickness data. The performance of the five meta-heuristic algorithms was evaluated using the COVID-19 symptoms dataset. The COVID-19 symptom dataset was created in accordance with WHO and the Indian ministry of health and family welfare criteria. The findings provide the average classification accuracy for each approach.

Corresponding Author:

Mohd Arfian Ismail
Faculty of Computing, Universiti Malaysia Pahang
26600 Pekan, Pahang, Malaysia
Email: arfian@ump.edu.my

1. INTRODUCTION

A newly found coronavirus, severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), has triggered a global pandemic of respiratory sickness dubbed coronavirus disease 2019 (COVID-19) [1]–[24]. COVID-19 appears to be more contagious than influenza and results in more serious diseases in certain people [5], [11], [15]. COVID-19 has a variable effect on individuals. COVID-19 infection has been associated with a wide variety of symptoms, ranging from mild to severe illness. Symptoms often manifest 2-14 days following viral contact. Fever, fatigue, a dry cough, difficulty breathing, sore throat, nasal congestion, runny nose, and diarrhoea are just a few of the pre-defined typical symptoms. These symptoms are based on world health organization (WHO) regulations [3], [12], [13], [19], [20], [25]. The severity of the sickness assists the medical team in isolating patients and providing appropriate health care.

It was found that using the fuzzy system is a suitable approach in identifying to identify the severity of the COVID-19 illness as it implements fuzzy logic and approximate reasoning [26], [27]. In addition, the fuzzy system also uses expert knowledge to ensure the system performs better. Fuzzy parameters are required to ensure the system able to work, and this process is called fuzzy modelling [26], [28]–[36]. Regrettably, when used to sophisticated problems such as engineering or medicine, the fuzzy systems architecture becomes complicated [26], [37]–[44]. This could occur as a result of the detection of numerous fuzzy parameters. A method for determining the suitable fuzzy parameters is required to address this issue [26], [33], [36], [45]–[61]. As a result, an automated procedure for identifying fuzzy parameters was developed [33], [36], [46], [62]–[66].

Fuzzy modelling is possible to be automated using metaheuristic algorithms. Metaheuristic algorithms are a type of computational intelligence paradigm that is particularly well-suited for solving complex optimization problems [29], [67]–[75]. Metaheuristic algorithms is potential to construct an automated method for determining the severity of a COVID-19 infection infected person based their symptoms [76]. Five metaheuristic algorithms were employed in this work to assess the severity of people's COVID-19 illness: differential evolution (DE), teaching-learning-based optimization (TLBO), particle swarm optimization (PSO), genetic algorithm (GA), and bat algorithm (BA). The next section describes these methods and illustrates them using their existing implementations.

2. META-HEURISTIC ALGORITHMS

2.1. Differential evolution

Differential evolution (DE) is a form of evolutionary algorithm invented by Storn and Price [77]. DE is a fast evolutionary algorithm (EA) that was developed for the purpose of solving optimization problems with real-valued parameters [73], [78]–[96]. DE enables the resolution of optimization problems by maintaining a population of candidate solutions. It generates new candidate solutions by merging existing solutions in the current generation to solve the optimization problem and retaining the candidate solution with the highest quality in the subsequent generation. DE is a three-step process that includes mutation crossover and selection [86], [97]–[100]. The flowchart in Figure 1 depicts the DE process.

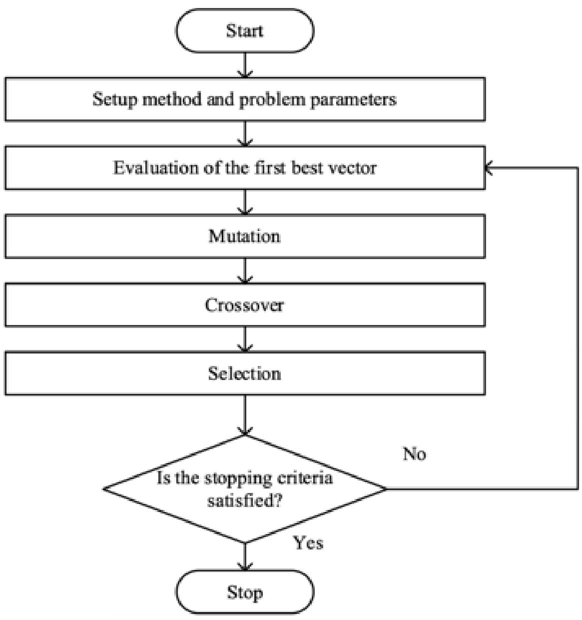


Figure 1. Flowchart of DE

2.2. Genetic algorithm

Genetic algorithm (GA) was introduced by John Holland in 1975 [101] mimics the natural concepts, which are genetic to represent the solution and selection, crossover, mutation to perform its operation. At each phase, GAs employ three distinct sorts of rules to generate the next generation from the present population: selection, crossover, and mutation [93], [95], [102]–[113]. Selection rules determine which individuals, referred to as parents, contribute to the population in the following generation. Crossover rules create the subsequent generation by combining the offspring of two parents. Individual parents are subjected to random mutations in order to produce children. The process of GA is depicted in flowchart form in Figure 2.

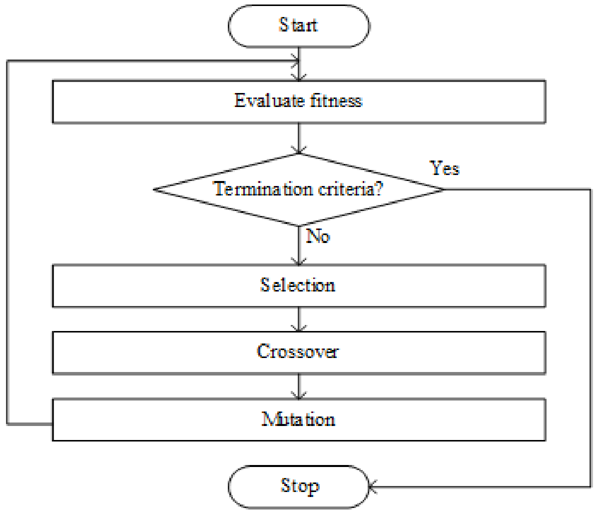


Figure 2. Flowchart of GA

2.3. Particle swarm optimization

Particle swarm optimization (PSO) is a form of swarm intelligence and swarm search that evolved from artificial life and evolutionary computation [114]–[124]. It is modelled after the social behaviour of flocking birds or schooling fish [124]. Since its inception in 1995, PSO has been successfully employed as a solution to a variety of function optimization problems or problems that able to be turned into function optimization problems [116]. Due to its lower memory needs and superior performance in offering solutions that are closer to the optimal for a variety of benchmarks and technical challenges such as computer vision [72], [93], [114], [120], [125]–[158]. PSO has become one of the most popular methods for tackling optimization problems [116][159]–[161]. The operation of PSO is depicted in flowchart form in Figure 3.

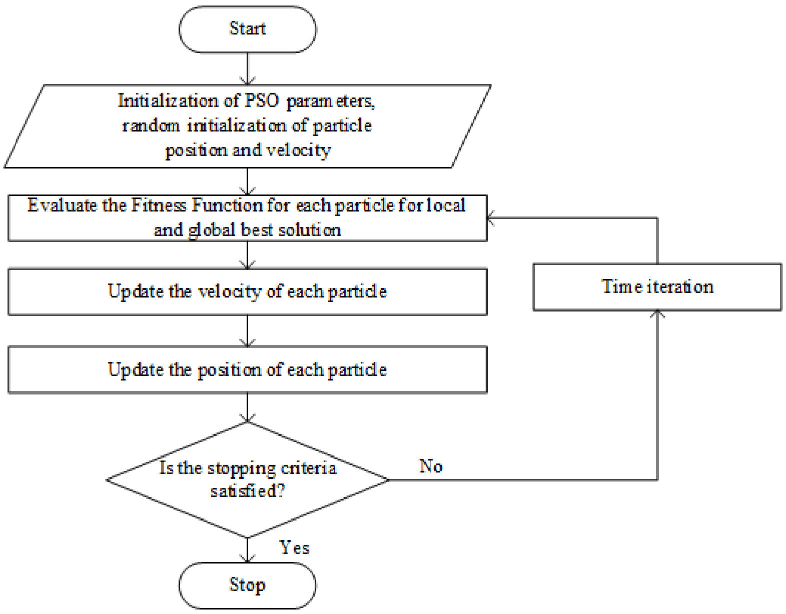


Figure 3. Flowchart of PSO algorithm

2.4. Teaching-learning-based optimization

The teacher-learning-based optimization (TLBO) algorithm is a novel form of metaheuristic algorithm motivated by the teaching-learning process. The TLBO method is a population-based optimization technique in which a group or class of students is considered a population. Each student is an embodiment of a potential solution to the problem. The numerous courses offered in class are interpreted as diverse design variables for the optimization problem, and students performance is interpreted as the fitness value of probable solutions to the optimization problem. The TLBO algorithm is separated into two phases: teacher and student [79], [162]–[170].

The teacher phase is the stage during which pupils learn directly from the teacher. According to the teaching-learning concept, the teacher is defined as the most experienced, knowledgeable, and highly educated member of society. The teacher makes an effort to increase pupils knowledge and to assist students in achieving high grades. Pupils, on the other hand, acquire knowledge and earn grades based on the quality of their teachers instruction and the quality of their classmates. Students acquire information throughout the student phase through mutual engagement. To improve knowledge, a student interacts randomly with other students in the class. If another student in the class has more knowledge than the former, the latter will teach the former something new. Thus, if the latter student is superior to the former, the former student is relegated to the latter. Otherwise, the former student will be separated from the latter. TLBO operates and processes in the manner depicted in Figure 4.

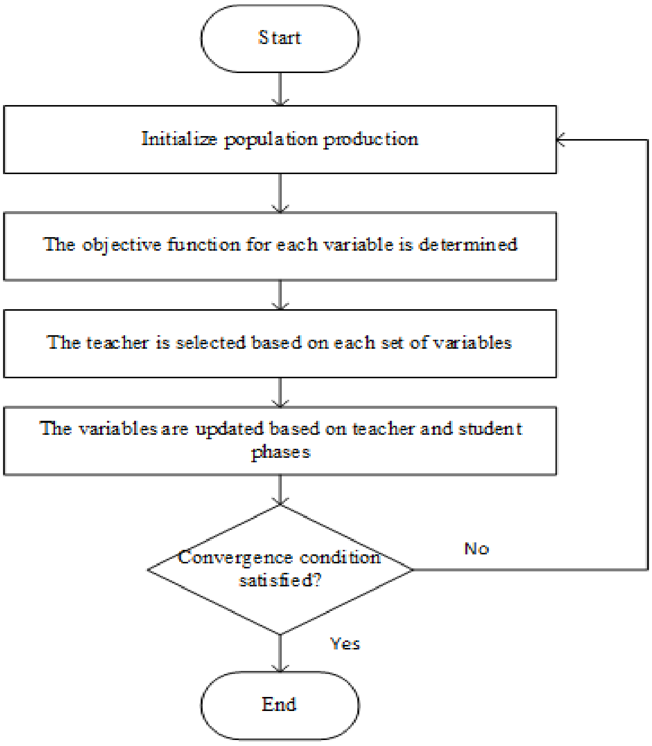


Figure 4. Flowchart of TLBO algorithm

2.5. Bat algorithm

The bat algorithm (BA) was introduced by Yang [171] in 2010 as a way for searching for global optimal solutions. BA has gained increased attention due to its simplicity, a small number of parameters, high robustness, and ease of implementation [144], [172]–[177]. The BA is designed to replicate bats echolocation activity. Bats generate a very loud sound pulse and listen for echoes in their environment. This signal varies according to the species of bat [178]. BA is founded on two tenets: i) all bats utilize echolocation and fly randomly in search of prey; and ii) all bats use echolocation. They distinguish between victim or food and impediment or background barriers; iii) bats utilize echolocation to determine distance. Bats able to automatically alter the wavelength of their outgoing pulses in response to their target. Figure 5 illustrates the steps required in BA.

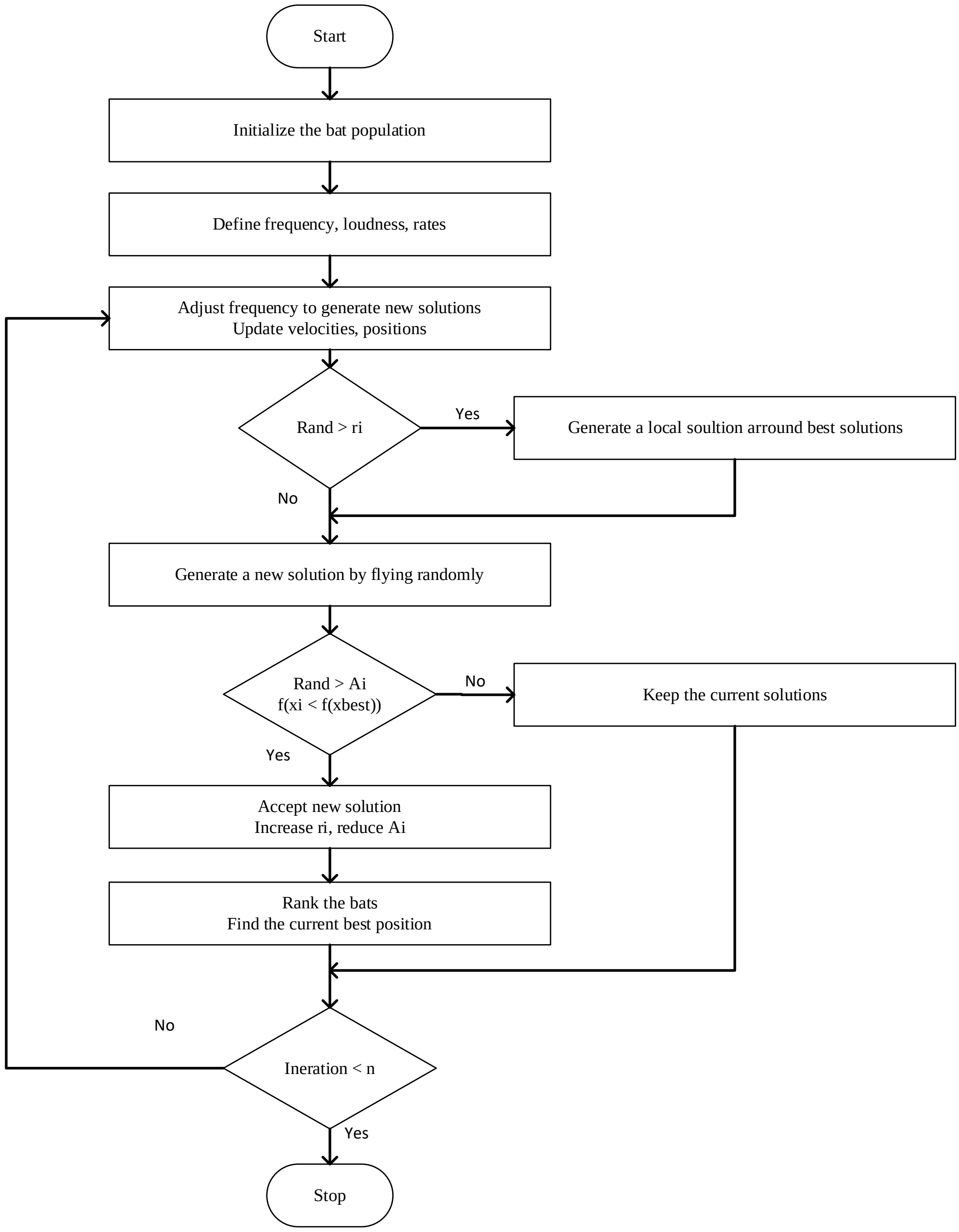


Figure 5. Flowchart of BAT algorithm

3. FUZZY META-HEURISTIC ALGORITHMS

This section discusses how meta-heuristic algorithms [29], [67]–[70], [73]–[75], [82], [90], [95], [109], [119], [160], [163], [176], [179]–[188] are included in the fuzzy modelling process. The implementation of fuzzy meta-heuristic algorithms is depicted in Figure 6. The initialization of the first population is the first process. The fuzzy parameters, including the fuzzy rules and fuzzy membership function, were produced randomly and represented by candidate solutions during this process. Figure 7 illustrates the representation of fuzzy rules and the fuzzy membership functions. Following that, the fuzzy rules and fuzzy membership functions are evaluated [189]. The fuzzy system classified data using fuzzy rules and the fuzzy membership functions. At this stage, proposed solutions were reviewed to see if they satisfied the termination requirement or not. If the termination requirement is met, the process terminates, and the optimal solution is found; and the process will continue when the termination condition is not achieved. The reproduction procedure seeks to increase the candidate solutions quality. The replication procedure is dependent on the meta-heuristic algorithm utilized in the fuzzy modelling process [29], [88], [180].

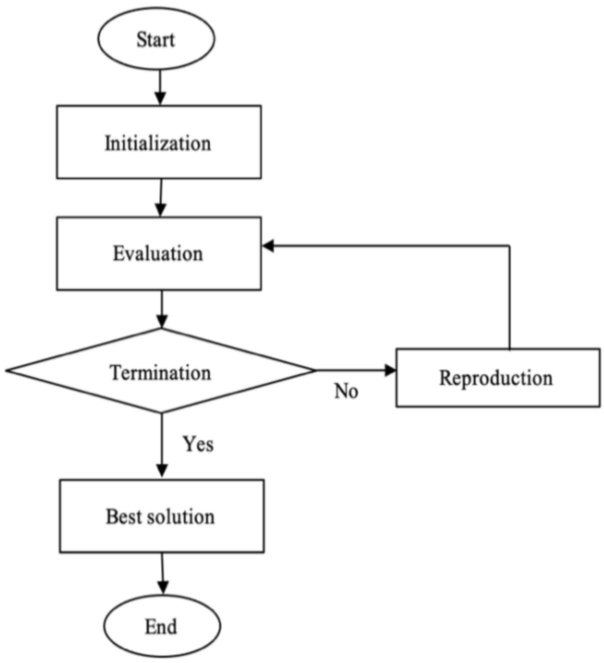


Figure 6. Flowchart of fuzzy meta-heuristic algorithm

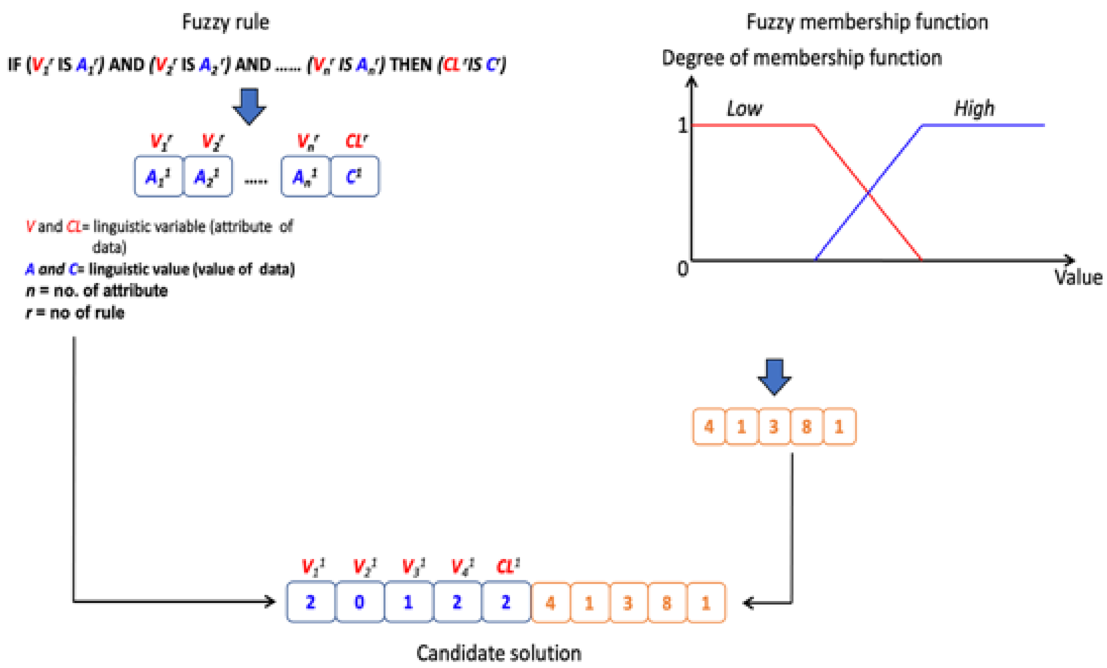


Figure 7. The representation of fuzzy rules and fuzzy membership function

4. EXPERIMENTAL DATA

In evaluating the performance of metaheuristic algorithms, the COVID-19 symptoms checker dataset was utilized. This dataset is available for download on the Kaggle website at https://www.kaggle.com/iamhungundji/covid19-symptoms-checker. This dataset contains information on COVID-19 classification (whether or not a person has the disease) based on a set of pre-defined standard symptoms. These symptoms are based on WHO and the Indian Ministry of Health and Family Welfare recommendations. The dataset contains significant variables that is utilized to determine whether or not someone has coronavirus illness. There are 316801 instances in the dataset.

5. RESULTS AND DISCUSSION

Following the collection of data, the performance of each machine-learning technique was examined. The various experiments results were presented and contrasted. The results revealed disparate values for various variables, including the number of correctly identified occurrences and the time required to execute the experiment. Table 1 gives the experiment's results. From the table, the fuzzy DE and fuzzy PSO algorithms performed well, with an average accuracy of 74.9 percent, followed by the fuzzy GA at 25.6 percent. Fuzzy TLBO and fuzzy BA had the lowest average accuracies at 25.1 percent each.

Table 1. The result of the meta-heuristic algorithm for fuzzy modelling in the classification of COVID-19

Method	Average accuracy (%)
Fuzzy DE	74.9
Fuzzy GA	25.6
Fuzzy PSO	74.9
Fuzzy TLBO	25.1
Fuzzy BA	25.1

Along with average accuracy, computational time was considered. Table 2 summarises the time required (in minutes) for each machine-learning approach to handle both datasets. According to the table, fuzzy DE outperformed the other approaches for both datasets. This could be due to the speed with which fuzzy DE performed the classification process, as it worked with the maximum margin, allowing for extremely low classification error [190]. It is robust and quick in classifying objects [191], [192]. Meanwhile, fuzzy TLBO required the most time to classify all datasets. This could be because the fuzzy TLBO's slow training procedure utilized a large amount of computer memory, making it time demanding [193]–[195]. The comparison of time expresses in Table 2.

Table 2. The comparison of time

Method	Computational time (sec)
Fuzzy DE	5.9771
Fuzzy GA	11.4683
Fuzzy PSO	6.5285
Fuzzy TLBO	146.7146
Fuzzy BA	14.5643

6. CONCLUSION

Machine learning is one of the methods that is suitable to be applied for classifying the severity of peoples COVID-19 illnesses. Five machine-learning algorithms were evaluated in this study: fuzzy DE, fuzzy GA, fuzzy PSO, fuzzy TLBO, and fuzzy BA. The benchmark dataset was created in accordance with WHO requirements. The data set included pre-defined standard symptoms such as fever, fatigue, dry coughing, trouble breathing, sore throat, nasal congestion, runny nose, and diarrhoea. Numerous experiments were conducted to show the ability of the methods. The performance criteria included the precision with which the data were classified correctly and the time required to conduct the experiment. As a result of the data, it is clear that fuzzy DE performs admirably in terms of average accuracy and classification time. As a result, the fuzzy DE fared the best in classifying the COVID-19 Symptoms Checker dataset.

ACKNOWLEDGEMENTS

The authors thank to Universiti Malaysia Pahang, Universiti Tun Hussein Onn, Universiti Teknologi Malaysia and Universitas Ahmad Dahlan for supporting this collaborative research in the present work. This work was supported by the Postgraduate Research Scheme (PGRS) vot No PGRS200396 from Universiti Malaysia Pahang and Universiti Teknologi Malaysia under Grant vot No. PY/2020/04173.

REFERENCES

[1] W. H. Organization, "Naming the coronavirus disease (COVID-19) and the virus that causes it," 2020. https://www.who.int/emergencies/diseases/novel-coronavirus-2019/technical-guidance/naming-the-coronavirus-disease-(covid-2019)-and-the-virus-that-causes-it.

[2] S. Krishnan, P. Magalingam, and R. Ibrahim, "Hybrid deep learning model using recurrent neural network and gated recurrent unit for heart disease prediction," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 6, pp. 5467–5476, 2021, doi:

10.11591/ijece.v11i6.pp5467-5476.
[3] N. N. Sari, M. N. Gani, R. A. Maharani Yusuf, and R. Firmando, "Telemedicine for silent hypoxia: Improving the reliability and accuracy of Max30100-based system," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 22, no. 3, pp. 1419–1426, 2021, doi: 10.11591/ijeecs.v22.i3.pp1419-1426.
[4] R. Alotaibi and A. Alghamdi, "Studying faculty members' readiness to use Shaqra University e-learning platform," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 22, no. 3, pp. 1556–1564, 2021, doi: 10.11591/ijeecs.v22.i3.pp1556-1564.
[5] N. A. Hidayat, P. Megantoro, A. Yurianta, A. Sofiah, S. A. Aldhama, and Y. A. Effendi, "The application of instrumentation system on a contactless robotic triage assistant to detect early transmission on a COVID-19 suspect," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 22, no. 3, pp. 1334–1344, 2021, doi: 10.11591/ijeecs.v22.i3.pp1334-1344.
[6] E. A. Mahareek, A. S. Desuky, and H. A. El-Zhni, "Simulated annealing for svm parameters optimization in student's performance prediction," *Bull. Electr. Eng. Informatics*, vol. 10, no. 3, pp. 1211–1219, 2021, doi: 10.11591/eei.v10i3.2855.
[7] T. Wahyuningrum, C. Kartiko, A. C. Wardhana, and T. G. Soares, "Revised web impact factor analysis of timor leste university website during covid-19 pandemic," *Bull. Electr. Eng. Informatics*, vol. 10, no. 3, pp. 1678–1686, 2021, doi: 10.11591/eei.v10i3.3034.
[8] N. Hayatin, K. M. Ghufron, and G. W. Wicaksono, "Summarization of COVID-19 news documents deep learning-based using transformer architecture," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 19, no. 3, pp. 754–761, 2021, doi: 10.12928/TELKOMNIKA.v19i3.18356.
[9] S. Kim and K. You, "Data analysis of financial burden index through KBO league FA pitcher's performance and contract amount size," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 3, pp. 2525–2562, 2021, doi: 10.11591/ijece.v11i3.pp2555-2562.
[10] S. K. Addagarla and A. Amalanathan, "e-SimNet: A visual similar product recommender system for E-commerce," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 22, no. 1, pp. 563–570, 2021, doi: 10.11591/ijeecs.v22.i1.pp563-570.
[11] "Recently employed engineering techniques to reduce the spread of COVID-19 (Corona Virus disease 2019): A review study," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 22, no. 1, pp. 277–286, 2021, doi: 10.11591/ijeecs.v22.i1.pp277-286.
[12] M. W. Hasan, "Covid-19 fever symptom detection based on IoT cloud," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 2, pp. 1823–1829, 2021, doi: 10.11591/ijece.v11i2.pp1823-1829.
[13] M. Al-Smadi, M. Hammad, Q. B. Baker, S. K. Tawalbeh, and S. A. Al-Zboon, "Transfer deep learning approach for detecting coronavirus disease in X-ray images," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 6, pp. 4999–5008, 2021, doi: 10.11591/ijece.v11i6.pp4999-5008.
[14] J. Hurtado and F. Reales, "A machine learning approach for the recognition of melanoma skin cancer on macroscopic images," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 19, no. 4, pp. 1357–1368, 2021, doi: 10.12928/TELKOMNIKA.v19i4.20292.
[15] P. Mookdarsanit and L. Mookdarsanit, "The covid-19 fake news detection in thai social texts," *Bull. Electr. Eng. Informatics*, vol. 10, no. 2, pp. 988–998, 2021, doi: 10.11591/eei.v10i2.2745.
[16] L. Rafea, A. Ahmed, and W. D. Abdullah, "Classification of a COVID-19 dataset by using labels created from clustering algorithms," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 21, no. 1, pp. 164–173, 2021, doi: 10.11591/ijeecs.v21.i1.pp164-173.
[17] D. C. Tran, "An open toolbox for generating map of actively confirmed sars-cov-2 or covid-19 cases in vietnam," *Bull. Electr. Eng. Informatics*, vol. 9, no. 6, pp. 2396–2403, 2020, doi: 10.11591/eei.v9i6.2621.
[18] N. H. Hamada and F. F. Kharbat, "P-norms of histogram of oriented gradients (p-HOG) for X-ray images," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 5, pp. 4423–4430, 2021, doi: 10.11591/ijece.v11i5.pp4423-4430.
[19] D. Sanin-Villa, O. D. Monsalve-Cifuentes, and J. S. Del Rio, "Early fever detection on COVID-19 infection using thermoelectric module generators," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 5, pp. 3828–3837, 2021, doi: 10.11591/ijece.v11i5.pp3828-3837.
[20] N. Alrefai and O. Ibrahim, "Deep learning for COVID-19 diagnosis based on chest X-ray images," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 5, pp. 4531–4541, 2021, doi: 10.11591/ijece.v11i5.pp4531-4541.
[21] A. M. Saleh, H. Y. Abuaddous, O. Enaizan, and F. Ghabban, "User experience assessment of a COVID-19 tracking mobile application (AMAN) in Jordan," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 23, no. 2, pp. 1120–1127, 2021, doi: 10.11591/ijeecs.v23.i2.pp1120-1127.
[22] A. Y. Gheni, H. A. Yousif, and Y. Y. Jusoh, "A critical success factors for software project manager in gvts within covid-19 pandemic," *Bull. Electr. Eng. Informatics*, vol. 10, no. 4, pp. 2293–2301, 2021, doi: 10.11591/eei.v10i4.3120.
[23] M. M. Khubrani and S. Alam, "A detailed review of blockchain-based applications for protection against pandemic like COVID-19," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 19, no. 4, pp. 1185–1196, 2021, doi: 10.12928/TELKOMNIKA.v19i4.18465.
[24] M. M. Singh, R. Frank, and W. M. N. Wan Zainon, "Cyber-criminology defense in pervasive environment: A study of cybercrimes in Malaysia," *Bull. Electr. Eng. Informatics*, vol. 10, no. 3, pp. 1658–1668, 2021, doi: 10.11591/eei.v10i3.3028.
[25] W. H. Organization, "Coronavirus Disease," 2021. https://www.who.int/news-room/q-a-detail/coronavirus-disease-covid-19.
[26] N. S. Nordin, M. A. Ismail, V. Mezhuyev, S. Kasim, M. S. Mohamad, and A. O. Ibrahim, "Fuzzy Modelling using Firefly Algorithm for Phishing Detection," *Adv. Sci. Technol. Eng. Syst.*, vol. 4, no. 6, pp. 291–296, 2019, doi: 10.25046/aj040637.
[27] K. Mahmoud *et al.*, "Prediction of the effects of environmental factors towards COVID-19 outbreak using AI-based models," *IAES Int. J. Artif. Intell.*, vol. 10, no. 1, pp. 35–42, 2021, doi: 10.11591/ijai.v10.i1.pp35-42.
[28] C.-Y. Fan, P.-C. Chang, J.-J. Lin, and J. C. Hsieh, "A hybrid model combining case-based reasoning and fuzzy decision tree for medical data classification," *Appl. Soft Comput.*, vol. 11, no. 1, pp. 632–644, Jan. 2011, doi: 10.1016/j.asoc.2009.12.023.
[29] N. S. Nordin *et al.*, "A comparative analysis of metaheuristic algorithms in fuzzy modelling for phishing attack detection," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 23, no. 2, pp. 1146–1158, 2021, doi: 10.11591/ijeecs.v23.i2.pp1146-1158.
[30] A. Karimi and T. J. Gandomani, "Software development effort estimation modeling using a combination of fuzzy-neural network and differential evolution algorithm," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 1, pp. 707–715, 2021, doi: 10.11591/ijece.v11i1.pp707-715.
[31] M. Z. Ismail, M. H. N. Talib, Z. Ibrahim, J. Mat Lazi, and Z. Rasin, "Experimental simplified rule of self tuning fuzzy logic-model reference adaptive speed controller for induction motor drive," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 20, no. 3, pp. 1653–1664, 2020, doi: 10.11591/ijeecs.v20.i3.pp1653-1664.
[32] N. N. C. Razali, N. A. Ghani, S. I. Hisham, S. Kasim, N. S. Widodo, and T. Sutikno, "Rainfall-runoff modelling using adaptive neuro-fuzzy inference system," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 17, no. 2, pp. 1117–1126, 2020, doi: 10.11591/IJEECS.V17.I2.PP1117-1126.
[33] L. B. Khuzyatova and L. A. Galiullin, "Optimization of parameters of neuro-fuzzy model," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 19, no. 1, pp. 229–232, 2020, doi: 10.11591/ijeecs.v19.i1.pp229-232.
[34] E. H. Karam, N. A. Al-Awad, and N. S. Abdul-Jaleel, "Design nonlinear model reference with fuzzy controller for nonlinear SISO second order systems," *Int. J. Electr. Comput. Eng.*, vol. 9, no. 4, pp. 2491–2502, 2019, doi: 10.11591/ijece.v9i4.pp2491-

2402.

[35] Y. S. Triana and A. Retnowardhani, “Enhance interval width of crime forecasting with ARIMA model-fuzzy alpha cut,” *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 17, no. 3, pp. 1193–1201, 2019, doi: 10.12928/TELKOMNIKA.v17i3.12233.

[36] L. B. Khuzyatova and L. A. Galiullin, “Optimization of parameters of neuro-fuzzy model,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 17, no. 3, pp. 1206–1209, 2019, doi: 10.11591/ijeecs.v17.i3.pp1206-1209.

[37] N. Azieta, M. Aseri, N. A. M. Aseri, M. A. Ismail, A. S. Fakharudin, and A. O. Ibrahim, “Review of The Meta-Heuristic Algorithms for Fuzzy Modeling in The Classification Problem,” *Int. J.*, vol. 9, no. 1.4, pp. 387–400, 2020, doi: 10.30534/ijatcse/2020/5691.42020.

[38] I. Alexander, C. Daniel, A. Paúl, and A.-A. Diego, “Performance analysis of optimized controllers with bio-inspired algorithms,” *Bull. Electr. Eng. Informatics*, vol. 9, no. 6, pp. 2507–2517, 2020, doi: 10.11591/eei.v9i6.2619.

[39] J. T. Santoso, M. C. Wibowo, B. Raharjo, and M. Mufadhol, “Gammu and kalkun for information services and sales based on information technology,” *Int. J. Electr. Comput. Eng.*, vol. 10, no. 2, pp. 2110–2116, 2020, doi: 10.11591/ijece.v10i2.pp2110-2116.

[40] K. A. R. Vargas, O. J. S. Parra, and M. J. Espitia Rico, “Temperature monitoring and forecast system in remote areas with 4.0G LTE mobile technologies,” *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 16, no. 5, pp. 1988–2004, 2018, doi: 10.12928/TELKOMNIKA.v16i5.10246.

[41] A. M. Rezaiee and A. Karimi, “A new dynamic intelligent model to determine reliability and trust of online banking by using fuzzy C-mean,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 4, no. 3, pp. 605–610, 2016, doi: 10.11591/ijeecs.v4.i3.pp605-610.

[42] B. D. Vighneshwari and R. Neela, “Insight on effectiveness of frequently exercised PQ classification techniques,” *Int. J. Electr. Comput. Eng.*, vol. 6, no. 5, pp. 2437–2446, 2016, doi: 10.11591/ijece.v6i5.10744.

[43] M. Outanoute, A. Lachhab, A. Ed-Dahhak, M. Guerbaoui, A. Selmani, and B. Bouchikhi, “Synthesis of an optimal dynamic regulator based on linear quadratic Gaussian (LQG) for the control of the relative humidity under experimental greenhouse,” *Int. J. Electr. Comput. Eng.*, vol. 6, no. 5, pp. 2262–2273, 2016, doi: 10.11591/ijece.v6i5.10470.

[44] T. Siraj Ahmed, S. Sao, and K. S. R. Anjaneyulu, “Microcontroller based stator resistance determination of induction motor on temperature variations,” *Int. J. Power Electron. Drive Syst.*, vol. 4, no. 3, pp. 356–362, 2014, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84901753705&partnerID=40&md5=3aed50b0095ad7f71666d07524d8a05a.

[45] K. M. Arun Prasad and U. Nair, “Intelligent fuzzy sliding mode controller based on FPGA for the speed control of a BLDC motor,” *Int. J. Power Electron. Drive Syst.*, vol. 11, no. 1, pp. 477–486, 2020, doi: 10.11591/ijpeds.v11.i1.pp477-486.

[46] T. Dewi, S. Nurmaini, P. Risma, Y. Oktarina, and M. Roriz, “Inverse kinematic analysis of 4 DOF pick and place arm robot manipulator using fuzzy logic controller,” *Int. J. Electr. Comput. Eng.*, vol. 10, no. 2, pp. 1376–1386, 2020, doi: 10.11591/ijece.v10i2.pp1376-1386.

[47] M. Irwanto, H. Alam, M. Masri, B. Ismail, W. Z. Leow, and Y. M. Irwan, “Solar energy density estimation using ANFIS based on daily maximum and minimum temperature,” *Int. J. Power Electron. Drive Syst.*, vol. 10, no. 4, pp. 2206–2213, 2019, doi: 10.11591/ijpeds.v10.i4.2206-2213.

[48] V. Kasinathan, A. Mustapha, T. S. Tong, M. F. C. A. Rani, and N. A. A. Rahman, “Heartbeats: Music recommendation system with fuzzy inference engine,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 16, no. 1, pp. 275–282, 2019, doi: 10.11591/ijeecs.v16.i1.pp275-282.

[49] F. Hashim, R. Mohamad, M. Kassim, S. I. Suliman, N. M. Anas, and A. Z. A. Bakar, “Implementation of embedded real-time monitoring temperature and humidity system,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 16, no. 1, pp. 184–190, 2019, doi: 10.11591/ijeecs.v16.i1.pp184-190.

[50] S. N. Syed Nasir, J. J. Jamian, and M. W. Mustafa, “Minimizing harmonic distortion impact cause by cs using meta heuristic technique,” *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 17, no. 4, pp. 1992–2000, 2019, doi: 10.12928/TELKOMNIKA.v17i4.12768.

[51] F. S. M. Alkhafaji, W. Z. Wan Hasan, M. M. Isa, and N. Sulaiman, “A response time reduction for DC motor controller using SISO technique,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 17, no. 2, pp. 895–906, 2019, doi: 10.11591/ijeecs.v17.i2.pp895-906.

[52] S. M. Khatami, M. Maadi, and R. Ramezani, “A clustering expert system using particle swarm optimization and k-means++ for journal recommendation to publish the papers,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 12, no. 2, pp. 814–823, 2018, doi: 10.11591/ijeecs.v12.i2.pp814-823.

[53] V.-D. Bui, T.-T. Nguyen, N.-H. Than, and D.-M. Nguyen, “A suitable structure to control the system of quad-rotor miniature aerial vehicles,” *Int. J. Power Electron. Drive Syst.*, vol. 9, no. 4, pp. 1634–1643, 2018, doi: 10.11591/ijpeds.v9n4.pp1634-1643.

[54] H. S. Hamad and A. H. Ali, “An adaptable different-levels cascaded H-bridge inverter analysis for PV grid-connected systems,” *Int. J. Power Electron. Drive Syst.*, vol. 9, no. 4, pp. 1755–1764, 2018, doi: 10.11591/ijpeds.v9n4.pp1755-1764.

[55] H. K. Omar, K. H. Jihad, and S. F. Hussein, “Comparative analysis of the essential cpu scheduling algorithms,” *Bull. Electr. Eng. Informatics*, vol. 10, no. 5, pp. 2742–2750, 2021, doi: 10.11591/eei.v10i5.2812.

[56] Q.-V. Ngo and T.-T. Nguyen, “The mppt algorithm combined with pitch angle control for the small-scale wind turbine in a wide speed range,” *Int. J. Power Electron. Drive Syst.*, vol. 12, no. 3, pp. 1482–1493, 2021, doi: 10.11591/ijpeds.v12.i3.pp1482-1493.

[57] A. H. M. Almawgani *et al.*, “Smart and efficient system for the detection of wrong cars parking,” *Bull. Electr. Eng. Informatics*, vol. 10, no. 4, pp. 1968–1978, 2021, doi: 10.11591/EEI.V10I4.2634.

[58] Y. Baba and M. Bouzi, “A study on modeling of a piezoelectric motor,” *Int. J. Power Electron. Drive Syst.*, vol. 12, no. 2, pp. 695–702, 2021, doi: 10.11591/ijpeds.v12.i2.pp695-702.

[59] M. Moutchou, A. Jbari, and Y. Abouelmahjoub, “Implementation of reduced induction machine fuzzy logic control, based on dspace-1104 r&d controller board,” *Int. J. Power Electron. Drive Syst.*, vol. 12, no. 2, pp. 1015–1023, 2021, doi: 10.11591/ijpeds.v12.i2.pp1015-1023.

[60] Z. Mekrini, S. Bri, J. Bouchnaif, and B. Bossoufi, “Experimental results of vector control for an asynchronous machine,” *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 18, no. 6, pp. 3285–3292, 2020, doi: 10.12928/TELKOMNIKA.v18i6.12707.

[61] A. A. Firdaus, R. T. Yunardi, E. I. Agustin, T. E. Putri, and D. O. Anggriawan, “Short-term photovoltaics power forecasting using Jordan recurrent neural network in Surabaya,” *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 18, no. 2, pp. 1089–1094, 2020, doi: 10.12928/TELKOMNIKA.v18i2.14816.

[62] N. S. Nordin, M. Ismail, and N. Omar, “Fuzzy modelling using butterfly optimization algorithm for phishing detection,” *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 1 Special Issue 5, pp. 355–360, Oct. 2020, doi: 10.30534/ijatcse/2020/5091.52020.

[63] S. Berhil, H. Benlahmar, and N. Labani, “A review paper on artificial intelligence at the service of human resources

management," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 18, no. 1, pp. 32–40, 2019, doi: 10.11591/ijeecs.v18.i1.pp32-40.
[64] A. H. Miry and G. A. Aramice, "Water monitoring and analytic based ThingSpeak," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 4, pp. 3588–3595, 2020, doi: 10.11591/ijece.v10i4.pp3588-3595.
[65] D. Kucherov, A. Kozub, O. Sushchenko, and R. Skrynkovskyy, "Stabilizing the spatial position of a quadrotor by the backstepping procedure," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 23, no. 2, pp. 1188–1199, 2021, doi: 10.11591/ijeecs.v23.i2.pp1188-1199.
[66] M. Díaz-Choque *et al.*, "Automated system for monitoring and control of the liquid wax production process," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 23, no. 2, pp. 782–790, 2021, doi: 10.11591/ijeecs.v23.i2.pp782-790.
[67] M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, "Chapter 10 - Metaheuristic Algorithms: A Comprehensive Review," in *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*, A. K. Sangaiah, M. Sheng, and Z. Zhang, Eds. Academic Press, 2018, pp. 185–231.
[68] M. Shehab, A. T. Khader, and M. A. Al-Betar, "A survey on applications and variants of the cuckoo search algorithm," *Appl. Soft Comput.*, 2017, doi: 10.1016/j.asoc.2017.02.034.
[69] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Inf. Sci. (Ny).*, vol. 237, pp. 82–117, 2013.
[70] J. B. Odili, "The dawn of metaheuristic algorithms," *Int. J. Softw. Eng. Comput. Syst.*, vol. 4, no. 2, pp. 49–61, 2018.
[71] S. Binitha and S. S. Sathya, "A Survey of Bio inspired Optimization Algorithms," *Int. J. Soft Comput. Eng.*, vol. 2, no. 2, pp. 137–151, 2012.
[72] S. Kadry, V. Rajinikanth, J. Koo, and B.-G. Kang, "Image multi-level-thresholding with Mayfly optimization," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 6, pp. 5420–5429, 2021, doi: 10.11591/ijece.v11i6.pp5420-5429.
[73] T. P. Van, V. Snášel, and T. T. Nguyen, "Antlion optimization algorithm for optimal non-smooth economic load dispatch," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 2, pp. 1187–1199, 2020, doi: 10.11591/ijece.v10i2.pp1187-1199.
[74] M. Al-Shabi, C. Ghenai, M. Bettayeb, F. F. Ahmad, and M. E. H. Assad, "Estimating pv models using multi-group salp swarm algorithm," *IAES Int. J. Artif. Intell.*, vol. 10, no. 2, pp. 398–406, 2021, doi: 10.11591/IJAI.V10.I2.PP398-406.
[75] A. V Vitianingsih, N. Suryana, and Z. Othman, "Spatial analysis model for traffic accident-prone roads classification: A proposed framework," *IAES Int. J. Artif. Intell.*, vol. 10, no. 2, pp. 365–373, 2021, doi: 10.11591/ijai.v10.i2.pp365-373.
[76] S. S. Mohamed Ali, A. H. Alsaeedi, D. Al-Shammary, H. H. Alsaeedi, and H. W. Abid, "Efficient intelligent system for diagnosis pneumonia (SARSCOVID19) in X-ray images empowered with initial clustering," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 22, no. 1, pp. 241–251, 2021, doi: 10.11591/ijeecs.v22.i1.pp241-251.
[77] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
[78] S.Poongothai *et al.*, "Application of Fuzzy Differential Evolutionary Algorithms in Biological Data Mining," *Int. J. Sci. Eng. Res.*, vol. 10, no. 1, pp. 1379–1382, 2019, [Online]. Available: http://www.ijser.org.
[79] S. R. Salkuti, "Short-term optimal hydro-thermal scheduling using clustered adaptive teaching learning based optimization," *Int. J. Electr. Comput. Eng.*, vol. 9, no. 5, pp. 3359–3365, 2019, doi: 10.11591/ijece.v9i5.pp3359-3365.
[80] N. F. B. Aswan, M. N. Abdullah, and A. H. A. Bakar, "A review of combined economic emission dispatch for optimal power dispatch with renewable energy," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 16, no. 1, pp. 33–40, 2019, doi: 10.11591/ijeecs.v16.i1.pp33-40.
[81] L. H. Pham, T. T. Nguyen, L. D. Pham, and N. H. Nguyen, "Stochastic fractal search based method for economic load dispatch," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 17, no. 5, pp. 2535–2546, 2019, doi: 10.12928/TELKOMNIKA.v17i5.12539.
[82] K. Rajesh and N. Visali, "Hybrid method for achieving Pareto front on economic emission dispatch," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 4, pp. 3358–3366, 2020, doi: 10.11591/ijece.v10i4.pp3358-3366.
[83] A. M. Abdulazeez, D. M. Hajy, D. Q. Zeebaree, and D. A. Zebari, "Robust watermarking scheme based LWT and SVD using artificial bee colony optimization," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 21, no. 2, pp. 1218–1229, 2020, doi: 10.11591/ijeecs.v21.i2.pp1218-1229.
[84] M. Belazzoug, A. Chanane, and K. Sebaa, "An efficient NSCE algorithm for multi-objective reactive power system compensation with UPFC," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 22, no. 2, pp. 40–51, 2020, doi: 10.11591/ijeecs.v22.i2.pp40-51.
[85] M. Iqbal, N. M. Nawi, and R. B. Mohamad, "An improved flower pollination solution for economic dispatch with valve point effect," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 22, no. 2, pp. 21–29, 2020, doi: 10.11591/ijeecs.v22.i2.pp21-29.
[86] S. K. Gachhayat and S. K. Dash, "Modified sub-gradient based combined objective technique and evolutionary programming approach for economic dispatch involving valve-point loading, enhanced prohibited zones and ramp rate constraints," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 5, pp. 5048–5057, 2020, doi: 10.11591/IJECE.V10I5.PP5048-5057.
[87] H. S. Maharana and S. K. Dash, "Dual objective multiconstraint swarm optimization based advanced economic load dispatch," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 3, pp. 1924–1932, 2021, doi: 10.11591/ijece.v11i3.pp1924-1932.
[88] B. B. Acharya, S. Dhakal, A. Bhattarai, and N. Bhattarai, "Pid speed control of dc motor using meta-heuristic algorithms," *Int. J. Power Electron. Drive Syst.*, vol. 12, no. 2, pp. 822–831, 2021, doi: 10.11591/ijpeds.v12.i2.pp822-831.
[89] T. M. Phan, P. T. Ha, T. L. Duong, and T. T. Nguyen, "Improved particle swarm optimization algorithms for economic load dispatch considering electric market," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 4, pp. 3918–3926, 2020, doi: 10.11591/ijece.v10i4.pp3918-3926.
[90] A. H. Halim and I. Ismail, "Tree physiology optimization in constrained optimization problem," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 16, no. 2, pp. 876–882, 2018, doi: 10.12928/TELKOMNIKA.v16i2.9021.
[91] A. Oukennou, A. Sandali, and S. Elmoumen, "Coordinated placement and setting of FACTS in electrical network based on Kalai-Smorodinsky bargaining solution and voltage deviation index," *Int. J. Electr. Comput. Eng.*, vol. 8, no. 6, pp. 4079–4088, 2018, doi: 10.11591/ijece.v8i6.pp.4079-4088.
[92] A. Azmi, S. M. Zali, M. N. Abdullah, M. F. N. Tajuddin, and S. R. A. Rahim, "The performance of COR optimization using different constraint handling strategies to solve ELD," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 17, no. 2, pp. 680–688, 2019, doi: 10.11591/ijeecs.v17.i2.pp680-688.
[93] N. Hashim and Z. Salam, "Critical evaluation of soft computing methods for maximum power point tracking algorithms of photovoltaic systems," *Int. J. Power Electron. Drive Syst.*, vol. 10, no. 1, pp. 548–561, 2019, doi: 10.11591/ijpeds.v10n1.pp548-561.
[94] I. Tahyudin and H. Nambo, "Improved optimization of numerical association rule mining using hybrid particle swarm optimization and cauchy distribution," *Int. J. Electr. Comput. Eng.*, vol. 9, no. 2, pp. 1359–1373, 2019, doi: 10.11591/ijece.v9i2.pp1359-1373.
[95] S. C. B. Semlali, M. E. Riffi, and F. Chebihi, "Parallel hybrid chicken swarm optimization for solving the quadratic assignment problem," *Int. J. Electr. Comput. Eng.*, vol. 9, no. 3, pp. 2064–2074, 2019, doi: 10.11591/ijece.v9i3.pp2064-2074.

[96] S. A. Yasear and K. R. Ku-Mahamud, "Non-dominated sorting Harris's hawk multi-objective optimizer based on reference point approach," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 15, no. 3, pp. 1603–1614, 2019, doi: 10.11591/ijeecs.v15.i3.pp1603-1614.
[97] K. Devika and G. Jeyakumar, "Theoretical analysis and empirical comparison of different population initialization techniques for evolutionary algorithms," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 12, no. 1, pp. 87–94, 2018, doi: 10.11591/ijeecs.v12.i1.pp87-94.
[98] K. Cabana, J. Candelo, R. Castillo, and E. De-La-Hoz-Franco, "Voltage sensitivity analysis to determine the optimal integration of distributed generation in distribution systems," *Int. J. Electr. Comput. Eng.*, vol. 9, no. 1, pp. 55–65, 2019, doi: 10.11591/ijece.v9i1.pp55-65.
[99] X. Tan and S.-Y. Shin, "Differential evolution algorithm of soft island model based on K-means clustering," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 19, no. 3, pp. 1548–1555, 2020, doi: 10.11591/ijeecs.v19.i3.pp1548-1555.
[100] A. Faris, I. Musirin, S. Jelani, S. A. Ismail, M. H. Mansor, and A. V Senthil Kumar, "Tap changer optimisation using embedded differential evolutionary programming technique for loss control in power system," *Bull. Electr. Eng. Informatics*, vol. 9, no. 6, pp. 2253–2260, 2020, doi: 10.11591/eei.v9i6.2505.
[101] J. R. Sampson, "Adaptation in natural and artificial systems (John H. Holland)." Society for Industrial and Applied Mathematics, 1976.
[102] A. A. K. Taher and S. M. Kadhim, "Improvement of genetic algorithm using artificial bee colony," *Bull. Electr. Eng. Informatics*, vol. 9, no. 5, pp. 2125–2133, 2020, doi: 10.11591/eei.v9i5.2233.
[103] Z. H. Ali, A. K. Hussein, H. K. Abass, and E. Fadel, "Extractive multi document summarization using harmony search algorithm," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 19, no. 1, pp. 89–95, 2021, doi: 10.12928/TELKOMNIKA.V19I1.15766.
[104] M. L. Mfenjou, A. A. A. Ari, A. N. Njoya, Kolyang, W. Abdou, and A. M. Gueroui, "A multi-objective evolutionary scheme for control points deployment in intelligent transportation systems," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 2, pp. 1641–1655, 2021, doi: 10.11591/ijece.v11i2.pp1641-1655.
[105] M. Chattoraj and U. R. Vinayakamurthy, "A self adaptive new crossover operator to improve the efficiency of the genetic algorithm to find the shortest path," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 23, no. 2, pp. 1011–1017, 2021, doi: 10.11591/ijeecs.v23.i2.pp1011-1017.
[106] S. S. Sadiq, A. M. Abdulazeez, and H. Haron, "Solving multi-objective master production schedule problem using memetic algorithm," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 18, no. 2, pp. 938–945, 2020, doi: 10.11591/ijeecs.v18.i2.pp938-945.
[107] Umar, Firdaus, A. Soeprijanto, and O. Penangsang, "Optimal expenditure and benefit cost based location, size and type of DGs in microgrids systems using adaptive real coded genetic algorithm," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 16, no. 1, pp. 10–17, 2018, doi: 10.12928/telkomnika.v16i1.6761.
[108] M. Yusoff and A. A. Othman, "Genetic algorithm with elitist-tournament for clashes-free slots of lecturer timetabling problem," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 12, no. 1, pp. 303–309, 2018, doi: 10.11591/ijeecs.v12.i1.pp303-309.
[109] K. Kamil, K. H. Chong, H. Hashim, and S. A. Shaaya, "A multiple mitosis genetic algorithm," *IAES Int. J. Artif. Intell.*, vol. 8, no. 3, pp. 252–258, 2019, doi: 10.11591/ijai.v8.i3.pp252-258.
[110] M. W. B. Peeyee, S. Abdul-Rahman, N. H. A. Hamid, and M. Zakaria, "Heuristic based model for groceries shopping navigator," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 16, no. 2, pp. 932–940, 2019, doi: 10.11591/ijeecs.v16.i2.pp932-940.
[111] D. L. Shanthi and K. Prasanna, "Energy efficient intelligent routing in WSN using dominant genetic algorithm," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 1, pp. 500–511, 2020, doi: 10.11591/ijece.v10i1.pp500-511.
[112] Z. A. Ali, S. A. Rasheed, and N. No'man Ali, "An enhanced hybrid genetic algorithm for solving traveling salesman problem," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 18, no. 2, pp. 1035–1039, 2020, doi: 10.11591/ijeecs.v18.i2.pp1035-1039.
[113] S. Masrom, M. Mohamad, S. M. Hatim, N. Baharun, N. Omar, and A. S. Abd. Rahman, "Different mutation and crossover set of genetic programming in an automated machine learning," *IAES Int. J. Artif. Intell.*, vol. 9, no. 3, pp. 402–408, 2020, doi: 10.11591/ijai.v9.i3.pp402-408.
[114] I. M. Khudher and Y. I. Ibrahim, "Swarm intelligent hyperdization biometric," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 18, no. 1, pp. 385–395, 2019, doi: 10.11591/ijeecs.v18.i1.pp385-395.
[115] K. Loubna, B. Bachir, and Z. Izeddine, "Ant colony optimization for optimal low-pass state variable filter sizing," *Int. J. Electr. Comput. Eng.*, vol. 8, no. 1, pp. 227–235, 2018, doi: 10.11591/ijece.v8i1.pp227-235.
[116] D. Tian, "EFPSO: An Effective Fuzzy Particle Swarm Optimization and Its Applications," 2018.
[117] M. Tuberquia and C. Hernandez, "New approaches in cognitive radios using evolutionary algorithms," *Int. J. Electr. Comput. Eng.*, vol. 8, no. 3, pp. 1636–1646, 2018, doi: 10.11591/ijece.v8i3.pp1636-1646.
[118] M. Sankaraiah, S. Suresh Reddy, and M. Vijaya Kumar, "GWO based optimal reactive power coordination of DFIG, ULTC and capacitors," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 11, no. 3, pp. 805–813, 2018, doi: 10.11591/ijeecs.v11.i3.pp805-813.
[119] M. A. A. Rahman, B. Ismail, K. Naidu, and M. K. Rahmat, "Review on population-based metaheuristic search techniques for optimal power flow," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 15, no. 1, pp. 373–381, 2019, doi: 10.11591/ijeecs.v15.i1.pp373-381.
[120] S. Ibrahim, N. E. A. Khalid, and M. Manaf, "CAPSOCA: Hybrid technique for nosologic segmentation of primary brain tumors," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 16, no. 1, pp. 267–274, 2019, doi: 10.11591/ijeecs.v16.i1.pp267-274.
[121] S. R. Salkuti, "Optimal location and sizing of DG and D-STATCOM in distribution networks," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 16, no. 3, pp. 1107–1114, 2019, doi: 10.11591/ijeecs.v16.i3.pp1107-1114.
[122] A. S. Alkhafaji, A. A. Al-Hayder, and A. S. Hassooni, "Hybrid IWOPSO optimization based marine engine rotational speed control automatic system," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 1, pp. 840–848, 2020, doi: 10.11591/ijece.v10i1.pp840-848.
[123] M. Yusoff, A. N. M. Basir, N. A. Kadir, and S. A. Bahari, "Evaluation of particle swarm optimization for strength determination of tropical wood polymer composite," *IAES Int. J. Artif. Intell.*, vol. 9, no. 2, pp. 364–370, 2020, doi: 10.11591/ijai.v9.i2.pp364-370.
[124] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, 1995, vol. 4, pp. 1942–1948.
[125] D. Li, Q. He, L. Chunli, and Y. Hongjie, "Local binary fitting segmentation by cooperative quantum particle optimization," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 15, no. 1, pp. 531–539, 2017, doi: 10.12928/TELKOMNIKA.v15i1.3159.
[126] O. P. Verma and N. Sharma, "Intensity preserving cast removal in color images using particle swarm optimization," *Int. J. Electr. Comput. Eng.*, vol. 7, no. 5, pp. 2581–2595, 2017, doi: 10.11591/ijece.v7i5.pp2581-2595.
[127] N. A. Md Ghani, S. B. A. Kamaruddin, N. M. Ramli, I. Musirin, and H. Hashim, "Modified BPNN via iterated least median squares, particle Swarm optimization and firefly algorithm," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 8, no. 3, pp. 779–786,

2017, doi: 10.11591/ijeecs.v8.i3.pp779-786.
[128] P. Hemavathi and A. N. Nandakumar, "Novel scheme for minimal iterative PSO algorithm for extending network lifetime of wireless sensor network," *Int. J. Electr. Comput. Eng.*, vol. 8, no. 2, pp. 1084–1091, 2018, doi: 10.11591/ijece.v8i2.pp1084-1091.
[129] M. R. M. Sapiee and K. A. M. Annuar, "Synchronous mobile Robots formation control," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 16, no. 3, pp. 1183–1192, 2018, doi: 10.12928/TELKOMNIKA.v16i3.8397.
[130] Z. Frijet, A. Zribi, and M. Chtourou, "An adaptive neural network controller based on PSO and gradient descent method for PMSM speed drive," *Int. J. Power Electron. Drive Syst.*, vol. 9, no. 3, pp. 1412–1422, 2018, doi: 10.11591/ijpeds.v9n3.pp1412-1422.
[131] M. Takruri, M. K. A. Mahmoud, and A. Al-Jumaily, "PSO-SVM hybrid system for melanoma detection from histo-pathological images," *Int. J. Electr. Comput. Eng.*, vol. 9, no. 4, pp. 2941–2949, 2019, doi: 10.11591/ijece.v9i4.pp2941-2949.
[132] G. Jayagopi and S. Pushpa, "On the classification of arrhythmia using supplementary features from tetrolet transforms," *Int. J. Electr. Comput. Eng.*, vol. 9, no. 6, pp. 5006–5015, 2019, doi: 10.11591/ijece.v9i6.pp5006-5015.
[133] H. Bouyghf, B. Benhala, and A. Raihani, "Analysis of the impact of metal thickness and geometric parameters on the quality factor-Q in integrated spiral inductors by means of artificial bee colony technique," *Int. J. Electr. Comput. Eng.*, vol. 9, no. 4, pp. 2918–2931, 2019, doi: 10.11591/ijece.v9i4.pp2918-2931.
[134] O. V Lee *et al.*, "A malicious URLs detection system using optimization and machine learning classifiers," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 17, no. 3, pp. 1210–1214, 2019, doi: 10.11591/ijeecs.v17.i3.pp1210-1214.
[135] A. Mishra and S. Dehuri, "Real-time online fingerprint image classification using adaptive hybrid techniques," *Int. J. Electr. Comput. Eng.*, vol. 9, no. 5, pp. 4372–4381, 2019, doi: 10.11591/ijece.v9i5.
[136] Y. Hendrawan, S. Widyaningtyas, and S. Sucipto, "Computer vision for purity, phenol, and pH detection of Luwak coffee green bean," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 17, no. 6, pp. 3073–3085, 2019, doi: 10.12928/TELKOMNIKA.v17i6.12689.
[137] V. Jaiswal, V. Sharma, and S. Varma, "MMFO: Modified moth flame optimization algorithm for region based RGB color image segmentation," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 1, pp. 196–204, 2020, doi: 10.11591/ijece.v10i1.pp196-204.
[138] I. K. Mohammed and A. I. Abdulla, "Elevation, pitch and travel axis stabilization of 3DOF helicopter with hybrid control system by GA-LQR based PID controller," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 2, pp. 1868–1884, 2020, doi: 10.11591/ijece.v10i2.pp1868-1884.
[139] M. Z. M. Tumari, A. F. Z. Abidin, A. S. R. A. Subki, A. W. A. Aziz, M. S. Saealal, and M. A. Ahmad, "Liquid slosh control by implementing model-free PID controller with derivative filter based on PSO," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 18, no. 2, pp. 750–758, 2020, doi: 10.11591/ijeecs.v18.i2.pp750-758.
[140] M. J. Fadhil, M. N. Hawas, and M. A. Naji, "Architecture neural network deep optimizing based on self organizing feature map algorithm," *Bull. Electr. Eng. Informatics*, vol. 9, no. 6, pp. 2538–2546, 2020, doi: 10.11591/eei.v9i6.1935.
[141] B. Madhu and G. Holi, "An optimal and secure watermarking system using SWT-SVD and PSO," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 18, no. 2, pp. 917–926, 2020, doi: 10.11591/IJEECS.V18.I2.PP917-926.
[142] M. S. Amiri, M. F. Ibrahim, and R. Ramli, "Optimal parameter estimation for a DC motor using genetic algorithm," *Int. J. Power Electron. Drive Syst.*, vol. 11, no. 2, pp. 1047–1054, 2020, doi: 10.11591/ijpeds.v11.i2.pp1047-1054.
[143] V. Sivasankarareddy and G. Sundari, "Survey on wireless sensor networks: Energy efficient optimization routing algorithms," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 19, no. 2, pp. 756–765, 2020, doi: 10.11591/ijeecs.v19.i2.pp756-765.
[144] N. H. Abbas, "Tuning of different controlling techniques for magnetic suspending system using an improved bat algorithm," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 3, pp. 2402–2415, 2020, doi: 10.11591/ijece.v10i3.pp2402-2415.
[145] S. B. Sabikan, S. W. Nawawi, and N. A. A. Aziz, "Modelling of time-to collision for unmanned aerial vehicle using particles swarm optimization," *IAES Int. J. Artif. Intell.*, vol. 9, no. 3, pp. 488–496, 2020, doi: 10.11591/ijai.v9.i3.pp488-496.
[146] H. Nurhadi, E. Apriliani, T. Herlambang, and D. Adzkiya, "Sliding mode control design for autonomous surface vehicle motion under the influence of environmental factor," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 5, pp. 4789–4797, 2020, doi: 10.11591/ijece.v10i5.pp4789-4797.
[147] A. M. Hasan and S. M. Rafaat, "Optimized formation control of multi-agent system using PSO algorithm," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 20, no. 3, pp. 1591–1600, 2020, doi: 10.11591/ijeecs.v20.i3.pp1591-1600.
[148] M. R. M. Romlay, S. F. Toha, A. M. Ibrahim, and I. Venkat, "Methodologies and evaluation of electronic travel aids for the visually impaired people: A review," *Bull. Electr. Eng. Informatics*, vol. 10, no. 3, pp. 1747–1758, 2021, doi: 10.11591/eei.v10i3.3055.
[149] K. S. Gaeid, A. F. Nashee, I. A. Ahmed, and M. H. Dekheel, "Robot control and kinematic analysis with 6DoF manipulator using direct kinematic method," *Bull. Electr. Eng. Informatics*, vol. 10, no. 1, pp. 70–78, 2021, doi: 10.11591/eei.v10i1.2482.
[150] X.-K. Dang, H.-N. Truong, V.-C. Nguyen, and T.-D.-A. Pham, "Applying convolutional neural networks for limited-memory application," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 19, no. 1, pp. 244–251, 2021, doi: 10.12928/TELKOMNIKA.V19I1.16232.
[151] A. F. Algamluoli and N. H. Abbas, "Speed controller design for three-phase induction motor based on dynamic adjustment grasshopper optimization algorithm," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 2, pp. 1143–1157, 2021, doi: 10.11591/ijece.v11i2.pp1143-1157.
[152] M. Yanto, S. Sanjaya, Yulasmi, D. Guswandi, and S. Arlis, "Implementation multiple linear regresion in neural network predict gold price," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 22, no. 3, pp. 1635–1642, 2021, doi: 10.11591/ijeecs.v22.i3.pp1635-1642.
[153] J. A. Jupin, T. Sutikno, M. A. Ismail, M. S. Mohamad, and S. Kasim, "Review of the machine learning methods in the classification of phishing attack," *Bull. Electr. Eng. Informatics*, vol. 8, no. 4, 2019, doi: 10.11591/eei.v8i4.1922.
[154] W. A. Jbara, "Ear biometric verification approach based on morphological and geometric invariants," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 20, no. 3, pp. 1479–1484, 2020, doi: 10.11591/ijeecs.v20.i3.pp1479-1484.
[155] V. Jaiswal, V. Sharma, and S. Varma, "An implementation of novel genetic based clustering algorithm for color image segmentation," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 17, no. 3, pp. 1461–1467, 2019, doi: 10.12928/TELKOMNIKA.v17i3.10072.
[156] Z. Feng and B. Zhang, "Fuzzy clustering image segmentation based on particle swarm optimization," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 13, no. 1, pp. 128–136, 2015, doi: 10.12928/TELKOMNIKA.v13i1.1269.
[157] S. Bhargava and A. Somkuwar, "Evaluation of noise exclusion of medical images using hybridization of partical swarm optimization and bivariate shrinkage methods," *Int. J. Electr. Comput. Eng.*, vol. 5, no. 3, pp. 421–428, 2015, doi: 10.11591/ijece.v5i3.pp421-428.
[158] D. K. Sambariya and S. Shrangi, "Optimal design of PID controller for load frequency control using harmony search algorithm," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 5, no. 1, pp. 19–32, 2017, doi: 10.11591/ijeecs.v5.i1.pp19-32.
[159] M. Rofiq *et al.*, "Integrating fuzzy logic and genetic algorithm for upwelling prediction in Maninjau Lake," *Telkomnika*

(Telecommunication Comput. Electron. Control., vol. 17, no. 1, pp. 226–234, 2019, doi: 10.12928/TELKOMNIKA.v17i1.11605.
[160] Y. K. Qawqzeh *et al.*, "Applying the big bang-big crunch metaheuristic to large-sized operational problems," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 3, pp. 2484–2502, 2020, doi: 10.11591/ijece.v10i3.pp2484-2502.
[161] O. M. Neda, "A new hybrid algorithm for solving distribution network reconfiguration under different load conditions," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 20, no. 3, pp. 1118–1127, 2020, doi: 10.11591/ijeecs.v20.i3.pp1118-1127.
[162] R. V Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-Learning-Based Optimization: An optimization method for continuous non-linear large scale problems," *Inf. Sci. (Ny).*, vol. 183, no. 1, pp. 1–15, Jan. 2012, doi: 10.1016/j.ins.2011.08.006.
[163] H. R. Ibraheem, Z. F. Hussain, S. M. Ali, M. Aljanabi, M. A. Mohammed, and T. Sutikno, "A new model for large dataset dimensionality reduction based on TLBO and LR," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 18, no. 3, pp. 1688–1694, 2020, doi: 10.12928/TELKOMNIKA.v18i3.13764.
[164] S. K. A. Attari, M. Bakhshipour, M. Shakarami, and F. Namdari, "A novel method based on teaching-learning-based optimization for recloser placement with load model consideration in distribution system," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 2, no. 1, pp. 1–10, 2016, doi: 10.11591/ijeecs.v2.i1.pp1-10.
[165] D. S. N. M. Rao and N. Kumar, "Comparisional investigation of load dispatch solutions with TLBO," *Int. J. Electr. Comput. Eng.*, vol. 7, no. 6, pp. 3246–3253, 2017, doi: 10.11591/ijece.v7i6.pp3246-3253.
[166] P. Chaudhury and H. K. Tripathy, "Optimising the parameters of a RBFN network for a teaching learning paradigm," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 15, no. 1, pp. 435–442, 2019, doi: 10.11591/ijeecs.v15.i1.pp435-442.
[167] S. R. Salkuti, "Solving optimal generation scheduling problem of Microgrid using teaching learning based optimization algorithm," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 17, no. 3, pp. 1632–1638, 2020, doi: 10.11591/ijeecs.v17.i3.pp1632-1638.
[168] M. Khorashadizade, M. Jouyban, and M. A. Oskoei, "A hybrid constructive algorithm incorporating teaching-learning based optimization for neural network training," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 4, pp. 3725–3733, 2020, doi: 10.11591/ijece.v10i4.pp3725-3733.
[169] S. R. Salkuti, "Power system state estimation using teaching learning-based optimization algorithm," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 18, no. 4, pp. 2125–2131, 2020, doi: 10.12928/TELKOMNIKA.v18i4.14159.
[170] M. Jouyban and M. Khorashadizade, "Using the modified k-mean algorithm with an improved teaching-learning-based optimization algorithm for feedforward neural network training," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 6, pp. 5277–5285, 2021, doi: 10.11591/ijece.v11i6.pp5277-5285.
[171] X. X. Ma and J. S. Wang, "Optimized Parameter Settings of Binary Bat Algorithm for Solving Function Optimization Problems," *J. Electr. Comput. Eng.*, vol. 2018, 2018, doi: 10.1155/2018/3847951.
[172] S. R. Borra, G. J. Reddy, and E. S. Reddy, "An efficient fingerprint identification using neural network and BAT algorithm," *Int. J. Electr. Comput. Eng.*, vol. 8, no. 2, pp. 1194–1213, 2018, doi: 10.11591/ijece.v8i2.pp1194-1213.
[173] Z. Boulouard, A. El Haddadi, F. Bouhafer, A. El Haddadi, L. Koutti, and B. Dousset, "Bat-cluster: A bat algorithm-based automated graph clustering approach," *Int. J. Electr. Comput. Eng.*, vol. 8, no. 2, pp. 1122–1130, 2018, doi: 10.11591/ijece.v8i2.pp1122-1130.
[174] R. B. Mohamed, M. M. Yusof, N. Wahid, N. Murli, and M. Othman, "Bat algorithm and k-means techniques for classification performance improvement," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 15, no. 3, pp. 1411–1418, 2019, doi: 10.11591/ijeecs.v15.i3.pp1411-1418.
[175] M. A. Faraj and A. M. Abbood, "Fractional order PID controller tuned by bat algorithm for robot trajectory control," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 21, no. 1, pp. 74–83, 2021, doi: 10.11591/ijeecs.v21.i1.pp74-83.
[176] A. N. Younis and F. M. Ramo, "A new parallel bat algorithm for musical note recognition," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 1, pp. 558–566, 2021, doi: 10.11591/ijece.v11i1.pp558-566.
[177] H. Setiadi, K. O. Jones, T. A. Nugroho, M. Abdillah, H. Trilaksana, and T. Amrillah, "Design of spark ignition engine speed control using bat algorithm," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 1, pp. 794–801, 2021, doi: 10.11591/ijece.v11i1.pp794-801.
[178] K. Premkumar and B. V Manikandan, "Bat algorithm optimized fuzzy PD based speed controller for brushless direct current motor," *Eng. Sci. Technol. an Int. J.*, vol. 19, no. 2, pp. 818–840, Jun. 2016, doi: 10.1016/j.jestch.2015.11.004.
[179] M. R. M. Ridzuan, E. E. Hassan, A. R. Abdullah, and A. F. A. Kadir, "Sustainable environmental economic dispatch optimization with hybrid metaheuristic modification," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 11, no. 1, pp. 161–168, 2018, doi: 10.11591/ijeecs.v11.i1.pp161-168.
[180] L. B. Amar and W. M. Jasim, "Hybrid metaheuristic approach for robot path planning in dynamic environment," *Bull. Electr. Eng. Informatics*, vol. 10, no. 4, pp. 2152–2162, 2021, doi: 10.11591/EEI.V10I4.2836.
[181] A. H. Alsaeedi, A. H. Aljanabi, M. E. Manna, and A. L. Albukhnefis, "A proactive metaheuristic model for optimizing weights of artificial neural network," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 20, no. 2, pp. 976–984, 2020, doi: 10.11591/ijeecs.v20.i2.pp976-984.
[182] L. A. Bewoor, V. C. Prakash, and S. U. Sapkal, "Comparative analysis of metaheuristic approaches for makespan minimization for no wait flow shop scheduling problem," *Int. J. Electr. Comput. Eng.*, vol. 7, no. 1, pp. 417–423, 2017, doi: 10.11591/ijece.v7i1.pp417-423.
[183] T. L. Duong, L. H. Pham, T. T. Nguyen, and T. T. Nguyen, "Application of a new constraint handling method for economic dispatch considering electric market," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 4. pp. 1542–1549, 2020, doi: 10.11591/eei.v9i4.2351.
[184] T. L. Duong, L. H. Pham, T. T. Nguyen, and T. T. Nguyen, "Application of a new constraint handling method for economic dispatch considering electric market," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 4. pp. 1542–1549, Aug. 2020, doi: 10.11591/eei.v9i4.2351.
[185] R. Patil, S. Tamane, S. A. Rawandale, and K. Patil, "A modified mayfly-SVM approach for early detection of type 2 diabetes mellitus," *Int. J. Electr. Comput. Eng.*, vol. 12, no. 1, pp. 524–533, 2022, doi: 10.11591/ijece.v12i1.pp524-533.
[186] M. M. Alhato and S. Bouallègue, "Thermal exchange optimization based control of a doubly fed induction generator in wind energy conversion systems," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 20, no. 3, pp. 1252–1260, 2020, doi: 10.11591/ijeecs.v20.i3.pp1252-1260.
[187] R. Sridhar, C. S. Boopathi, D. Das, S. Agrawal, and H. Choubisa, "An ingenious invasive weed optimization (IWO) aided maximum power tracking for partially shaded photovoltaic array," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 15, no. 2, pp. 543–553, 2019, doi: 10.11591/ijeecs.v15.i2.pp543-553.
[188] W. N. A.-D. Abed, O. A. Imran, and I. S. Fatah, "Automatic generation control based whale optimization algorithm," *Int. J. Electr. Comput. Eng.*, vol. 9, no. 6, pp. 4516–4523, 2019, doi: 10.11591/ijece.v9i6.pp4516-4523.
[189] A. Taieb and A. Ferdjouni, "A new design of fuzzy logic controller optimized by PSO-SCSO applied to SFO-DTC induction

motor drive," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 6, pp. 5813–5823, 2020, doi: 10.11591/ijece.v10i6.pp5813-5823.

[190] S. T. Suganthi, D. Devaraj, and S. H. Thilagar, "An improved differential evolution algorithm for congestion management considering voltage stability," *Indian J. Sci. Technol.*, vol. 8, no. 24, 2015, doi: 10.17485/ijst/2015/v8i24/80243.

[191] N. Karaboga and B. Cetinkaya, "Performance comparison of genetic and differential evolution algorithms for digital FIR filter design," in *International Conference on Advances in Information Systems*, 2004, pp. 482–488.

[192] a. D. Lilla, M. a. Khan, and P. Barendse, "Comparison of Differential Evolution and Genetic Algorithm in the design of permanent magnet Generators," *2013 IEEE Int. Conf. Ind. Technol.*, pp. 266–271, 2013, doi: 10.1109/ICIT.2013.6505683.

[193] A. García-Monzó, H. Migallón, A. Jimeno-Morenilla, J.-L. Sánchez-Romero, H. Rico, and R. V. Rao, "Efficient Subpopulation Based Parallel TLBO Optimization Algorithms," *Electronics*, vol. 8, no. 1, p. 19, 2019.

[194] S. Talatahari, V. Goodarzimehr, and N. Taghizadieh, "Hybrid Teaching-Learning-Based Optimization and Harmony Search for Optimum Design of Space Trusses," *J. Optim. Ind. Eng.*, vol. 13, no. 1, pp. 177–194, 2020, doi: 10.22094/joie.2019.1866904.1649.

[195] Z. Wu and R. Xue, "A Cyclical Non-Linear Inertia-Weighted Teaching–Learning-Based Optimization Algorithm," *Algorithms*, vol. 12, no. 5, p. 94, May 2019, doi: 10.3390/a12050094.

BIOGRAPHIES OF AUTHORS

Nur Azieta Mohamad Aseri is an admin staff of the Center Computing an Informatic, Universiti Malaysia Kelantan, Malaysia. She has experience in developing Postgraduate System for a local university by utilizing Agile Methodology using Java Spring Framework for back-end development and angular for front end development. His research interests include the field of Meta-Heuristic Algorithms, Fuzzy Modeling, Agile methodology, Java framework, and entrepreneur. Her key strengths include: working in a team, always punctual, committed with task given, innovative and creative, and can work in a stressful environment. She can be contacted at email: azieta.ma@umk.edu.my.

Mohd Arfian Ismail is a Senior Lecturer at the Faculty of Computer Systems and Software Engineering in University Malaysia Pahang, Malaysia. He received B. Sc, M. Sc. and PhD degree in Computer Science from Universiti Teknologi Malaysia (UTM) in 2008, 2011 and 2016, respectively. His current research interests are in the areas of machine learning methods and optimization method. He can be contacted at email: arfian@ump.edu.my.

Abdul Sahli Fakharudin is a lecturer in the Faculty of Computing, College of Computing and Applied Sciences, Universiti Malaysia Pahang. His research interest mainly in artificial neural network and machine learning and currently a member of Soft Computing and Intelligence Systems (SPINT) research group. He can be contacted at email: sahli@mp.edu.my.

Ashraf Osman Ibrahim obtained BSc and MSc in Computer Science from Al Neelain University and University of Khartoum, Sudan respectively. He did his Ph.D. in Computer Science, Universiti Teknologi Malaysia (UTM), 2015. His research interest includes: computational intelligence, soft computing, artificial neural networks, evolutionary algorithms, multi-objective evolutionary algorithms, machine learning, Pattern Classification and data mining. He can be contacted at email: ashrafosman2@gmail.com.

Shahreen Kasim is currently an Associate Professor in Department of Security Information and Web Technology, Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia. Her areas of interest include bioinformatics, soft computing, data mining, web and mobile application. She can be contacted at email: shahreen@uthm.edu.my.

Noor Hidayah Zakaria is a senior lecturer in the Faculty of Engineering, Universiti Teknologi Malaysia, Johor, Malaysia. Her areas of interest include neural netwok, fuzzy logic, ANFIS, soft computing, machine learning, applied artificial intelligence, social welfare and urban poverty. She can be contacted at email: noorhidayah.z@utm.my.

Tole Sutikno is a lecturer in Electrical Engineering Department at the Universitas Ahmad Dahlan (UAD), Yogyakarta, Indonesia. He received his B.Eng., M.Eng. and Ph.D. degrees in Electrical Engineering from Universitas Diponegoro, Universitas Gadjah Mada and Universiti Teknologi Malaysia, in 1999, 2004 and 2016, respectively. He has been an Associate Professor in UAD, Yogyakarta, Indonesia since 2008. He is currently an Editor-in-Chief of the TELKOMNIKA and the Head of the Embedded Systems and Power Electronics Research Group. His research interests include the field of digital design, industrial applications, industrial electronics, industrial informatics, power electronics, motor drives, renewable energy, FPGA applications, embedded system, artificial intelligence, intelligent control, information technology and digital library. He can be contacted at email: tole@ee.uad.ac.id.

IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 11, No. 1, March 2022, pp. 65~80
ISSN: 2252-8938, DOI: 10.11591/ijai.v11.i1.pp65-80

A deep learning-based multimodal biometric system using score fusion

Chahreddine Medjahed[1], Abdellatif Rahmoun[2], Christophe Charrier[3], Freha Mezzoudj[4]
[1]EEDIS Laboratory, Department of Computer Science, University of Djillali Liabes, Sidi Bel-Abbes, Algeria
[2]Department of Computer Science, ESI-SBA, University of Djillali Liabes, Sidi Bel-Abbes, Algeria
[3]GREYC Laboratory, Department of Multimedia and Internet, University of Caen Normandie, Caen, France
[4]Department of Computer Science, University Hassiba Benbouali of Chlef, Chlef, Algeria

Article Info

Article history:

Received Jul 8, 2021
Revised Nov 27, 2021
Accepted Dec 10, 2021

Keywords:

Biometric identification system
Multimodal biometric system
Deep learning
Identication

ABSTRACT

Recent trends in artificial intelligence tools-based biometrics have overwhelming attention to security matters. The hybrid approaches are motivated by the fact that they combine mutual strengths and they overcome their limitations. Such approaches are being applied to the fields of biomedical engineering. A biometric system uses behavioural or physiological characteristics to identify an individual. The fusion of two or more of these biometric unique characteristics contributes to improving the security and overcomes the drawbacks of unimodal biometric-based security systems. This work proposes efficent multimodal biometric systems based on matching score concatenation fusion of face, left and right palm prints. Multimodal biometric identification systems using convolutional neural networks (CNN) and k-nearest neighbors (KNN) are proposed and trained to recognize and identify individuals using multi-modal biometrics scores. Some popular biometrics benchmarks such as FEI face dataset and IITD palm print database are used as raw data to train the biometric systems to design a strong and secure verification/identification system. Experiments are performed on noisy datasets to evaluate the performance of the proposed model in extreme scenarios. Computer simulation results show that the CNN and KNN multi-modal biometric system outperforms most of the most popular up to date biometric verification techniques.

Corresponding Author:

Chahreddine Medjahed
EEDIS Laboratory, Department of Computer Science, University of Djillali Liabes
Sidi Bel-Abbes, 22000, Algeria
Email: chahreddine.medjahed@univ-sba.dz

1. INTRODUCTION

With the growth of intelligence artificial systems and e-technologies nowadays, personal biometric authentication has become an essential demanded technic: widely used in airports, buildings, mobile phones, identity cards and so on. The use of biometrics data is essential for learning powerful recognition systems. Many physiological traits (such as face, iris, fingerprint, palm-print, hand geometry, ear.) or behavioural ones (such as gait, signature, voice) are used to identify a person. These characteristics will not be lost or forgotten and can be used to distinguish one individual from another. The fusion of two or more of these characteristics contributes to improving the security and showing high performance and remedying the limits and the disadvantages of the unimodal biometric systems.

Face detection [1] task has the goal to detect all the human's faces in an image or sequence of images. Also, face identification (or recognition) [2], [3] system have the goal to detect a face in an image and then using classifiers or matching algorithms to identify or recognize who the face belongs to. However, face analysis [4] is the technique to examine an image and extract information, such as age, sex, complexion, emotion and so on.

Face identification is useful in variety of daily life areas such as healthcare system, authentification operation and so on. Face recognition is a convenient technic because it is easy to collect faces data without active cooperation of the person and faces data are more representative and discriminant for recognition. However, other biometric features, can be used to recognize individual such as palm print, fingerprint, gaits, signature, speech and so on.

Recently, palm print [5] has become one of the most notable biometric recognition systems and it has received interest of researchers. Many advantages led to use this trait such as less distortion, rich features and high accuracy. The principal lines, ridges and wrinkles in structure of palm print are stable all through the life of a person.

In general, there exist two types of the biometric systems: unimodal biometric systems and multimodal ones. Fisrstly, for unimodal biometric systems, one trait is used to identify a person. These systems can encounter different degradations and limitations such as lack of distinctiveness of the biometric trait or nonuniversality, noisy sensor data and so on. As a solution to these kinds of problems, multimodal biometric systems are created using many biometrics traits. This fusion reduces the risk of any spoofing or faking of other identities.

According to the literature, for multimodal system, the different traits are fused at one of these levels: data-sensor, feature-extraction, matching-score and decision levels. Recently, researchers are more interested by the fusion at matching-score level because of its better recognition accuracy compared to the other levels. According to [6], "the score level fusion is commonly preferred in multimodal biometric systems because matching scores contain sufficient information to distinguish between genuine and impostor cases".

This paper introduces and compares many unimodal and multimodal biometric systems for human identification. The authors present **s**trong multimodal biometric systems with deep learned and fuzzed scores' of three traits: face, left and right palm prints. First, the score of each modality is obtained using convolutional neural network (CNN) then, the fusion of scores helps to perform fusion at this level. The fusion of these modalities is implemented on Score Level using concatenation strategy. Second, k-nearest neighbors (KNN), the machine learning algorithm which remains a strong and a successful algorithm [7], [8] is used for the classification step. For more accurate evaluations and challenging situations, different kinds of biometric data are used: clear and noisy ones. Some variations in rotation and adding noises introduce large changes in faces and palms' images.

The rest of this paper is organized: an overview of previous works about multimodal biometric systems is presented in section 2. Section 3 summerises the techniques of deep learning neural networks used for scores learning and some machine learning tools dedicated for the classification. Section 4 describes the methodology of the used approach. Section 5 explores the experimental results. Section 6 concludes the work conducted and proposes some future works.

2. RELATED WORKS

Several works have demonstrated that a multi-modal biometric system can surpass some of inconveniencs of unimodal biometric system [9]. Many studies have suggested that by using information from multiple biometric traits, better performance can be achieved. In [10], Ross and Govindarajan have proposed multimodal biometric systems based on fusion of face and hand at feature level. Three different scenarios were developed. Firstly, a fusion of principal components analysis (PCA) and linear discriminant analysis (LDA), the principal components analysis and linear discriminant analysis algorithms respectively, face's coefficients was used. Second, a fusion of LDA coefficients which respesent the three channels of the face image: the red, green and blue was used. Finaly, fusion of face and hand traits was presented.

In [11], the authors proposed a fusion technique based on a discrete cosine transform (DCT) algorithm. A fused feature vector of face and palmprint data was constructed. The identification is done using gaussian mixture models (noted gaussian mixture model (GMM)). The proposed method produces good recognition rates when evaluated on FERET-PolyU and ORL-PolyU databases.

In [12], multimodal biometric was implemented based on fusion of retina, fingerprint and finger vein at feature level. The techniques such as blood vessel extraction, minutia extraction and maximum curvature were used to extract the useful features. The fuzed features are encrypted using the asymmetric public-key cryptosystem algorithm of rivest shamir adleman (RSA) and compared to a stored template to authenticate the person. The use of the RSA algorithm improves the baseline multimodal biometric's performance.

In [13], multimodal biometric systems with fusing the face, the palm print at different levels, sensor level, feature level, score level and decision level were introduced. The proposed systems were evaluated on the available publically PolyU and AR datasets for the palm print and face respectively. The result of this search showed the best perfomance is obtained with the score level fusion using sum rule with an accuracy of 97.5%.

In [14], the authors introduced a Multimodal biometric recognition system by combining face and both left iris and right iris. For face trait, the features were extracted with deep belief network (DBN). By applying CNN for each trait, the scores obtained were fused at two different levels: rank level and score level. Many databases were used to realize this work such as the facial recognition technology (FERET) database, SDUMLA HMT and CASIA V1.0.

In [15], the authors proposed multi-biometric systems for human verification using CNN to fuse iris and face traits on feature and score levels. They utilized the very deep CNN called VGG16 [16] to extract features from images. The recognition step is based only on the features without using any image detection techniques. The experimentations were conducted on the multimodal biometric database SDUMLA-HMT.

In our case, the main objective is to evaluate the performance of unimodal and multimodal biometric systems. As multimodalities, we use the fusion of the face, the right palm print and the left palm print traits at score level. The proposed models are based on deep learning models for feature extraction and machine learning tools for classification task, as illustrated in Figure 1. The evaluations of the proposed approach are done using clear and noisy and rotated data.

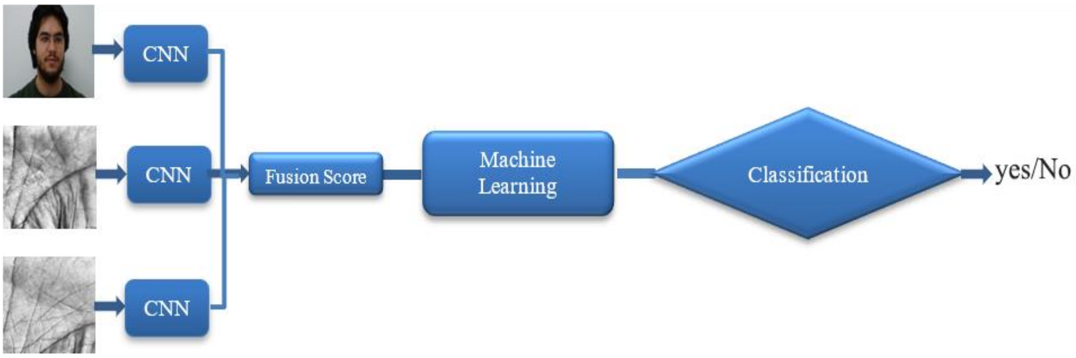


Figure 1. Pipeline tasks of our proposed methodology

3. MACHINE LEARNING APPROACHES

The main useful tools for our biometric systems are described. Three steps are involved is: (i) data pre-processing, (ii) feature extraction using the deep learning algorithms and (iii) training and testing identification person models using Machine learning algorithms.

3.1. Pre-processing

Differences in aging, occlusion, facial expressions, noises and poses faces' images constitute complex challenges for face recognition systems. In general, it is crucial, before any biometric recognition, to apply the face alignment, which contribute to detect the face area and to remove the background. Also, many technics are considered as image pre-processing and are used to enhance the quality of the data and facilitate the recognition task such as alignment face (or palm print), normalisation and de-noising.

However, other types of technics, such as deformation, scaling, rotation, changing colors, adding noises and so on, are applied on the original images for the data augmentation. In our case, we use some of these technics, such as adding noises and applying rotation, to decrease the images quality. Our goal is to obtain more challenging data as we can find in difficult or critical real situations.

The external disturbance such as environmental conditions during data acquisition or the quality of the sensing elements themselves can cause noise [17]. In this paper, we explore two types of noises: salt-and-pepper noise and gaussian noise. Also, rotations with different degrees are applied to the intial used data.

3.2. Convolutional neural networks

CNN are popular tools in the field of deep learning. Their robustness is due to their flexible architecture and their ability to extract features from raw data. They are successfully used in image classification [18], objects detection, Speech recognition and language modeling [19].

To accelerate the modeling and avoid the expensive computation and decrease the over-fitting due to the lack of the labeled data in some fields, many studies tuned and used deep pre-trained models (e.g,, AlexNet [20], VGG [16], GoogleNet [21], Resnet [22] and so on), as shown in Table 1. For image recognition task, the CNN input is an image with red, green and blue (RGB) channels and the output is the prediction of the image's category. These CNNs, mentioned above, are pre-trained on the dataset ImageNet [23] which is a dataset for computer vision research with more than 14 million of images.

In general, to train and test a CNN, series of convolution, pooling and fully connected layers are applied, followed by Softmax function to classify the data. These operations are the basic building blocks of every CNN. The kernel trick help to transform nonlinear case to linear one.The kernel size is choosen according to the variation in the lacation of the input information [24]. The *inception* [21] technic helps to have filters with multiple sizes operating on the same level**.** The *Dropout* [25] is used for the neural network regularization, which helps to reducing interdependent learning amongst the neurons.

Table 1. Comparison of some pre-trained CNNs

	AlexNet [20]	VGGNet [16]	GoogleNet [21]	RestNet [22]
#layers (convolutional + fully connected)	5+3	13/16+3	21+1	151+1
Kernel size	11, 5, 3	3	7, 1, 3, 5	7, 1, 3, 5
Data Aug.	+	+	+	+
Inception [21]	-	-	+	-
Dropout [25]	+	+	+	+

3.2.1. Convolutional layer

The convolutional layer is used to extract discriminative features from the images. "This bloc contains a set of convolution kernels (called filters). They are convolved with the input and generate a "feature map" [25]. Mathematically, the convolution procedure can be expressed using the (1):

$$y_{in} = f(\sum_{j=0}^{n} w_j\, x_j) \quad (1)$$

where *x* is an input value from the image, *w* is the weight value from the filter, the pixel number is noted by j. The function f is an activation function. The rectified activation function (ReLU) is widely used in Deep Learning. It replaces negative values with zero, according to the (2) as shown in Figure 2, where z is the convolutional layer output [26]:

$$f(u) = max(\,0, u) \quad (2)$$

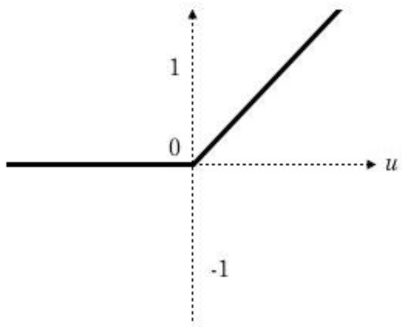


Figure 2. The ReLU function

3.2.2. Pooling layer

The second operation after convolution in the CNN is the Pooling. The pooling operation is helpful for acquiring a reduced component portrayal, which is invariant to direct changes in object scale, pose, and translation in an image [25]. Two kind of pooling operation are widely used: max pooling and average pooling. Max pooling compute the maximum element of the selection. It is most used type because it is fast to calculate and allows to effectively simplifying the image. For the average pooling, we calculate the average of the selection as shown in Figure 3.

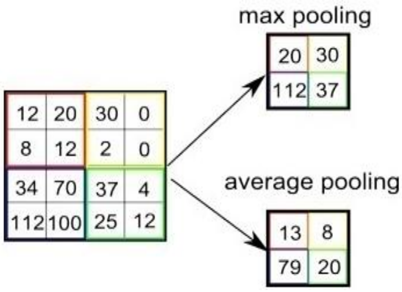


Figure 3. Pooling operations

3.2.3. Fully connected layer

The fully-connected layers are in the CNN's top. To facilite the classification task, it is necessary to convert the outputs of these previous fully-connected layers to probabilities. The *softmax function* is able to calculate them using the (3), where *m* is the class, *n* is the maximum classes' number, the output *y* is computed using the (4), where *x* is the feature vector of the data sample and *w* respresent the weight vector. The softmax classifier output which is a score vector represent a set of probabilities according to the different classes [26]:

$$soft\,max(\,i) = \frac{e^{y_i}}{\sum_{m=1}^{n} e^{y_m}} \tag{3}$$

$$y_{out} = \sum_{l=0}^{k} w_l\, x_l \tag{4}$$

3.3. Training convolution neural network

For training the CNN, "a loss function is used to estimate the quality of the prediction. This function quantifies the difference between the prediction made by the model and the correct output [25]. Training CNN is finding the best parameters of the network to reduce this function. There exist many types of loss function, such as: mean squared error, cross entropy loss and hinge loss. The type fuction must be choosen according to the traited problem. Gradient descent is the optimization algorithm employed to minimize the error by computing the gradient required for updating network parameter values.

3.3.1. AlexNet

AlexNet [20] is the first successful CNN for big data. It has a similar architecture to the original LeNet but it is deeper and wider CNN model. The architecture of AlexNet as shown in Figure 4 contains eight layers, five convolutions layer with max pooling and three fully connected layers. There are 60 million learning parameters and 650,000 neurons. AlexNet is the first CNN that uses ReLU activation function. The input of this CNN is RGB image with a size of 227×227×3.

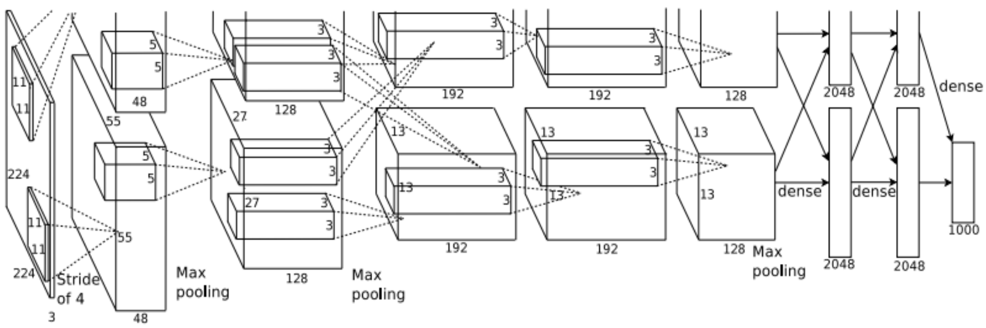


Figure 4. The architecture of AlexNet [20]

3.3.2. GoogleNet

In 2014, GoogleNet [21] has achieved the best result in ImageNet large scale visual recognition challenge (ILSVRC), the ImageNet large scale visual recognition challenge. Googlenet uses fewer parameters than the CNN AlexNet. GoogleNet implements Inception modules with the aim of optimizing the usage of computing resources within the network. The idea is to apply parallel pooling and convolutions operations with different kernel sizes and to concatenate the resulting feature maps before going to the next layer. GoogleNet has in total 22 layers and it uses an average pooling. The input of this CNN is RGB image with a size of 224×224×3 as shown in Figure 5.

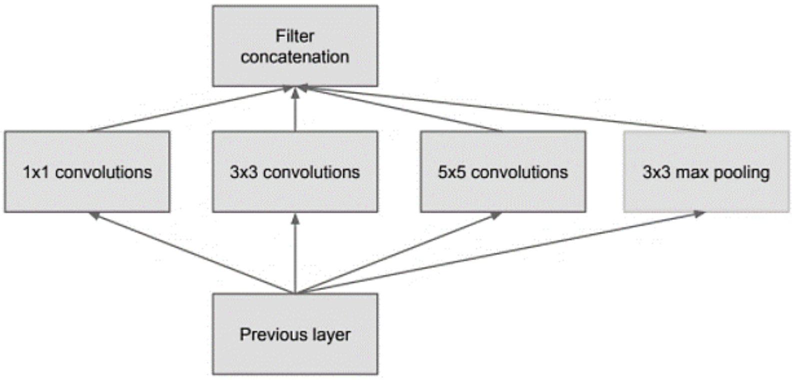


Figure 5. Simple scheme of an inception block as proposed by [21]

3.3.3. ResNet

In 2015, the Microsoft's residual network ResNet [22] has achieved the best result in ILSVRC, the ImageNet large scale visual recognition challenge. It was proposed with a residual learning block. Resnet overcome the problem of vanishing gradient and it is developed with different layers 18, 34, 50, and 101. The residual network architecture's remarkable feature is the identity skips connections within the residual blocks, which enables very deep CNN architectures to be trained easily. The residual network consists of several residual blocks which are stacked on top of each other [25], as illustrated in Figure 6.

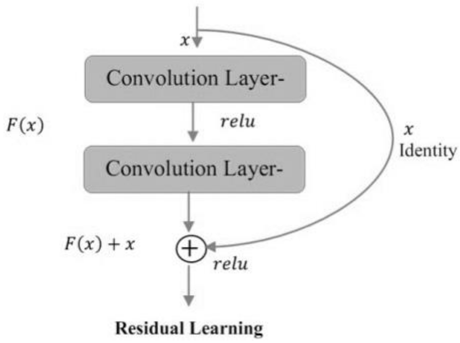


Figure 6. ResNet residual learning building block

3.4. Machine learning algorithms

Machine learning methods [27] are important tools for researchers, scientists and students in a wide range of areas. Traditionally, different techniques like k-nearest neighbors algorithm and support vector machines are used for the face recognition tasks [28]. These methods are based on hand crafted data representation such as detection of regions of interest and feature extraction. Among the feature's extraction methods: eigenfaces local and binary patterns. Are used. However, in our case, we use these machine learning methods to classify the scores obtained by the deep CNNs.

3.4.1. Naive Bayes

One straightforward source of classifier based on probability computation is the famous naïve Bayes classifier. There are many variants of this algorithm but all focus on the strong and naïve independence assumption between the features. "The naïve Bayes assumption is helpful when the dimensionality D of the input space is high, making density estimation in the full D-dimensional space more challenging". This supervised learning algorithm uses the famous Bayes theorem [24], [27].

3.4.2. Support vector machines (SVM)

SVM work on induction principle, called structural risk minimization, which targets to minimize an upper bound on the expected generalization error. The SVM uses the concept of mathematical planes, called maximum-margin hyperplanes, to distinguish between the different classes. It draws a plane between two classes. The SVM training consists on trying to maximize the distance of this plane from both classes using the concept of support vectors, which are the outermost points of each class. This margin is drawn explicitly in the case of a linear classification" [28], [29]. Also, inorder to find the hyperplane, the SVM uses the kernel trick with nonlinear classifications to transform nonlinear case to linear one. Also, the SVM was, first formulated for binary classification, and the extension to multi-classes is useful [27].

3.4.3. AdaBoost

In 1996, Freund and Schapire have developed AdaBoost, for adaptive boosting, which is an algorithm for combining many simple weak classifiers to obtain a strong classifier using a linear combination. It is a popular algorithm of machine learning that has the advantages of being quick in term of speed, easy to be programmed, simple in operation and there is no need to adjust parameters except for the number of iterations. AdaBoost algorithm generates a collection of bad learners by maintaining a weight over training data and adjusting them to each "weak period". The weights of the training samples misclassified by current poor learners will be increased while the weight of the correctly identified samples will be reduced [27].

3.4.4. Subspace discriminant

Subspace discriminant [27], [30] has been abundantly studied in data mining and pattern recognition. It is often combined and improved by the LDA which provid low-dimension for the discriminant subspace. Many studies have been performed to investigate the impact on the effectiveness on classification success in the ensemble learning of different subspacing, weighting and resampling techniques. Subspace discriminant model uses a random subspace algorithm to construct an ensemble of discriminant classifiers [25].

3.4.5. K-nearest neighbors

The supervised machine learning algorithm k-nearest neighbors (KNN) is based exclusively on the choice of classification metric. It is non-parametric, k must be fixed, and it is based on training data. The algorithm allows making a classification without making a hypothesis on the function $\mathrm{y} = \mathrm{f}(\mathrm{x}_1, \mathrm{x}_2, \dots, \mathrm{x}_\mathrm{p})$ which links the dependent variable to the independent variables.

The generalized distance between two variables is calculated using (5):

$$L_q = \left(\sum_{i,j=1}^{k} \left|x_i - x_{j\neq i}\right|^q\right)^{\frac{1}{q}} \tag{5}$$

when q=2, it is referred to euclidean distance and manhattan distance. The nearest neighbor is the variable with the shortest distance possible [7], [8], [27].

4. THE PROPOSED APPROACH DEEP LEARNING-BASED MULTIMODAL BIOMETRIC SYSTEM USING SCORE FUSION (DLMBS)

This section proposes a DLMBS. Firstly, we must identify which type of CNN is the best fit for such types of biometric data: face, left palm and left palm. These will be trained separately (or eventually simultaneously depending on the type of machine) up to feature layers at the score level (feature vectors). Then, score vectors will be fused to construct a multi-modal feature score. A separate experiment will be conducted to come up with the most performing way to combine such scores (linear combination, arithmetic averaging, and concatenation). This will be an input to a CNN that performs a final classification.

Other experiments will be done to test several machine learning (ML) classifiers. According to the best fit, we choose the best algorithm to construct the hybrid person identification system. The hybrid deep learning (DL), CNN, and ML models are based multi-modal scores. We notice that all these experiments are conducted using clean data.

With a similar scenario, we will test the effect of simulated noisy and oriented data on the proposed models. Two kinds of noises are introduced on the initial clean data. Also, some geometrical deformations are applied on the clean data. These simulated challenging situations help to test the robustness of the DLMBS performance.

4.1. Preliminary experiment

We create three separate unimodal biometric systems, based on respectively face, left palm, and right palm. Each of these biometric systems uses different types of CNN; ie; Alexnet, Googlenet and Resnet-18 neural networks respectively. These are also trained separately using standard datasets: FEI Face Dataset [31] and IITD Palm print Database [32] as shown in Figure 7.

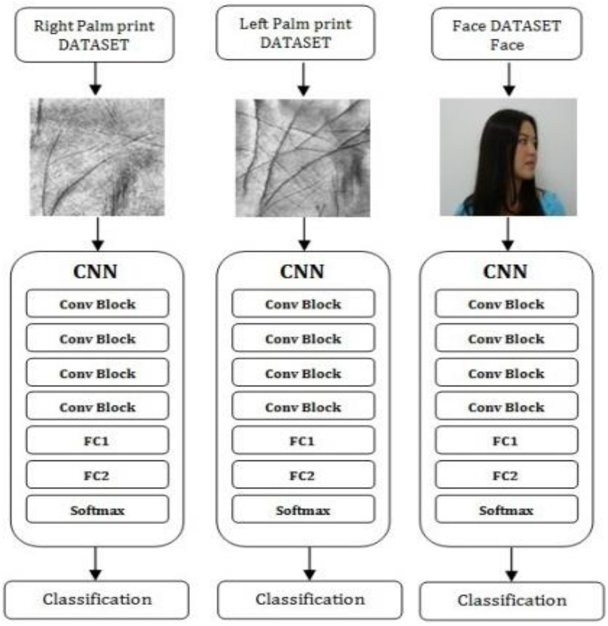


Figure 7. Part 1-unimodal biometric CNN models

4.1.1. Biometrics datasets

a) FEI Face dataset

The Brazilian FEI face database present a set of face images for 100 men and 100 women (200 individuals) that are students and staff of FEI laboratory between 19 and 40 years old. Each person has 14 images. Each image is with 640×480 pixels. All images are in color with different position of head, frontal pose and the head turning from left to right. Variations in illumination and head poses introduce large changes in images [31] as shown in Figure 8.

Figure 8. Sample of faces from FEI dataset with different head poses

b) IITD palm print V1 database

The IITD Palm print V1 database [32] as shown in Figure 9 is a hand database that contains a set of hand images with 800×600 pixels for 230 individuals that are students and staff at IIT Delhi campus, with 12-57 years old. Six or seven images from each subject, for each of the left and right palmprint, are acquired in different hand pose. Apart from the original images, there are also automatically cropped 150×150 pixels and normalized palm print images.

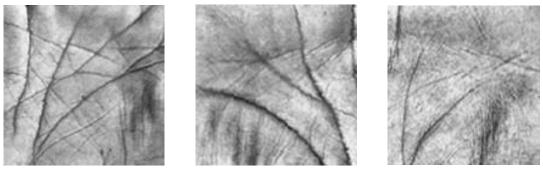


Figure 9. Sample of cropped Images from IITD Palm print V1 Dataset: each palm consists on principal lines, wrinkles and epidermal ridges

In this paper, we choose to use a subset that contains 140 subjects' faces from FEI face database and 140 hands' subjects from IITD palm print V1 database. Each subject has five different images from the three modalities (face, left palm print and right palm print) for training purposes. The training/testing ratio will be 80-20% respectively.

Thus, input matrices to CNNs will be of dimension (140×5) for each, and the values will be normalized to 1. We assign rows of the left palm prints and rows of the right palm prints to the corresponding rows of the face matrix. For sake of experimental convenience, we assume that every row (face, left palm, right palm) belongs to the same person, even though the two datasets FEI Face and IITD Palm print are of two different populations.

4.1.2. Training CNNs for separate modalities

In this section; we make an image classification for each modality; each modality is trained independently. All faces and palm print images are resized to 227×227, for AlexNet, 224×224; for GoogleNet and for Resnet 18. Table 2 shows the results for the unimodal identification biometric systems. We notice that the Resnet 18 neural network performs best for the face biometric system and the left and right palm print respectively. The resNet is successfully used in many fields and these results coincided with the literature [33], [34]. Apparently, ResNet 18 neural networks give the best accuracy rates for all three modalities.

Table 2. Results for unimodal systems of face and palm prints

Modalities	CNN	Time of Classification [s]	Accuracy Rate [%]
Face	AlexNet	10.26	99.28
	GoogleNet	25.29	97.14
	ResNet18	28.48	100
Left Palm Print	AlexNet	30.81	92.14
	GoogleNet	23.21	85.71
	ResNet18	30.10	95.00
Right Palm Print	AlexNet	3.96	87.14
	GoogleNet	15.26	86.43
	ResNet18	13.87	95.71

4.2. Training multi-modal biometric system (clean data)

The multimodal biometric system is evaluated by combining the face and the palmprints traits at score level. Preliminary experiments show that a concatenation as a fusion technique performs better than other types of combinations. The principal of the proposed person identification models is illustrated in the Figure 10. Here also, the two datasets FEI face and IITD Palm print V1 are used for the three CNNs training. The obtained scores are fused subsequently, and then classified with different types of ML classifiers. The Table 3 summarizes the most important evaluation results of the conducted experiments.

We observe that the fusion of two (or the three) biometric traits (face and Palm prints' scores using Resnet 18), as shown in Table 3, gives the best performance. The classification using Machine Learning algorithms such as SVM or naïve Bayes gives weakest results comparing to the results obtained by KNN, Adaboost and Subspace discriminant. Moreover, the central processing unit (CPU) processing time required by KNN for the classification step is very short. Furthermore, Resnet 18 neural network associated with KNN performs best for a multi-modal biometric system.

4.3. Training multi-modal biometric system (noisy data)

In this section, we will simulated and evaluated the effect of environment disturbance on the images during the acquisition process. The diversity of the angle during the acquisition of the image or the orientation of the capture devices or the low-quality sureveillance camera can affect the images quality.

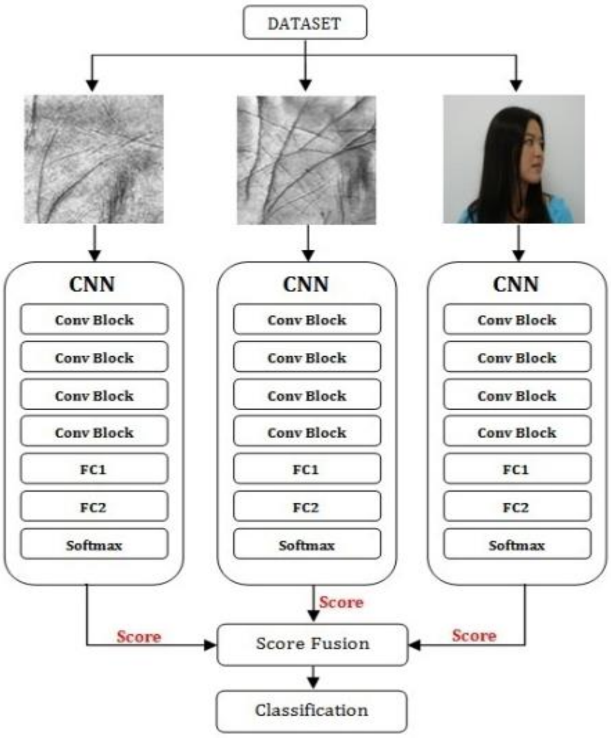


Figure 10. Multimodal biometric CNN model

Table 3. Results for multimodal systems of face, left and right palm prints (score level)

Modalities	Method	Time of Classification[s]	Accuracy Rate [%]
Face + Left Palm Print	Naive Bayes	59.959397	92.86
	SVM	1095.948599	78.57
	KNN	7.082602	100
	Adaboost	14.037205	100
	Subspace discriminant	44.535800	100
Face + Right Palm Print	Naive Bayes	71.17050	89.28
	SVM	1154.731083	77.86
	KNN	4.250831	100
	Adaboost	16.730207	100
	Subspace discriminant	59.260625	100
Left Palm Print + Right Palm Print	Naive Bayes	54.982823	81.43
	SVM	859.992817	57.14
	KNN	3.924279	100
	Adaboost	12.968442	93.57
	Subspace discriminant	41.010611	99.28
Face + Left Palm Print + Right Palm Print	Naive Bayes	77.295227	92.14
	SVM	902.416240	83.57
	KNN	5.248682	100
	Adaboost	13.430381	100
	Subspace discriminant	49.546649	100

4.3.1. The effect of noisy data on biometric systems

"Noise is a random variation of color information. It can affect the original signal and decrease it quality. Some external disturbances can be the cause such as: environmental conditions during image acquisition and the quality of the sensing elements themselves [17]. In order to simulate noisy data, we generate two kinds of noises: the Gaussian noise and the salt-and-pepper noise.

a) Salt and pepper noise

Salt-and-pepper noise in the images is due to faulty memory locations in hardware, malfunctioning pixels in camera and so on. The salt-and-pepper noise is also known as impulse noise, data drop noise or binary noise. Also, this type of noise can seen in the transmission of data and it appears as black dots on white background and white dots on a black one, as shown in Figure 11 [17].

Figure 11. Samples of face images with salt and pepper noise

b) Gaussian noise

Gaussian noise is also known as normal noise or white noise. Gaussian noise is caused by the discrete nature of warm object radiation and thermal atom vibration. [20]. The associated gaussian density function is given using the (6), also see the Figure 12:

$$P(z) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(z-\mu)^2}{2\sigma^2}} \quad (6)$$

where, the gray level is represented by z, the mean value is noted by μ, the standard deviation and the variation are noted by σ and σ^2respectively.

Figure 12. Samples of face images with gaussian noise

Two different noises are added to the data, salt-and-pepper and gaussian noises; gradually for face, left palm print and right palm print separately, then we combined these traits in score level with different possible scenarios. Figure 13 shows clearly that face is more resistant to noise than palm prints for the salt-and-pepper noise. The similar results are obtained with the gaussian noise.

Also, we compared the multi-modal biometric system (fused scores) versus the models trained on the data with the both types of noises gaussian and salt and pepper. We use the three modalities in our experiments. According to the obtained results of the accuracy, it is clear that combining face, left and right palm prints give a very accurate verification biometric system. We conclude that CNN and KNN model is robust and isn't badly affected by noise.

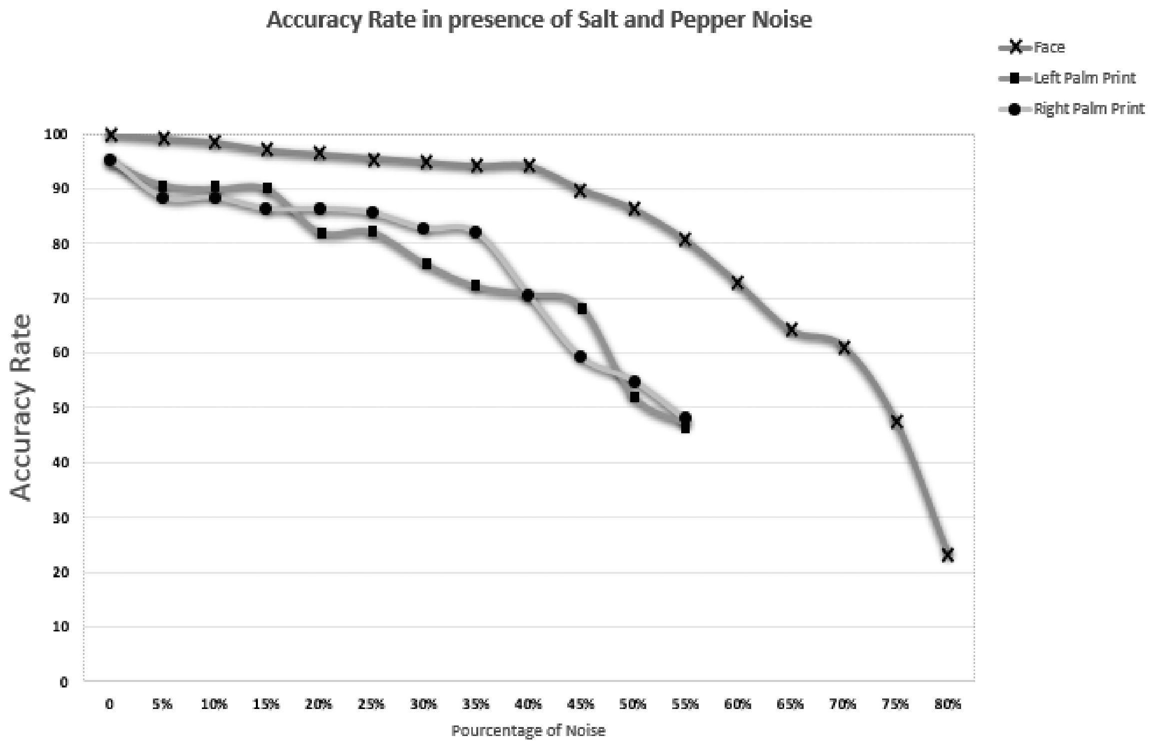


Figure 13. Accuracy rate in presence of salt and pepper noise

4.3.2. The effect of the geometrical deformation of images on biometric systems

We expose the experiments and their important results for image classification with different angle of rotation such as 0°, 30°, 45°, 60°, and 90°. We have generated multiple training images using rotation techniques from a training image. The principle is to use CNNs to analyse the classification performance on several variants of data as shown in Figure 14. The new simulated data involve both novel training and testing of the models. The Figure 15 shows verification accuracy for uni-modal systems using data with different degrees of rotation.

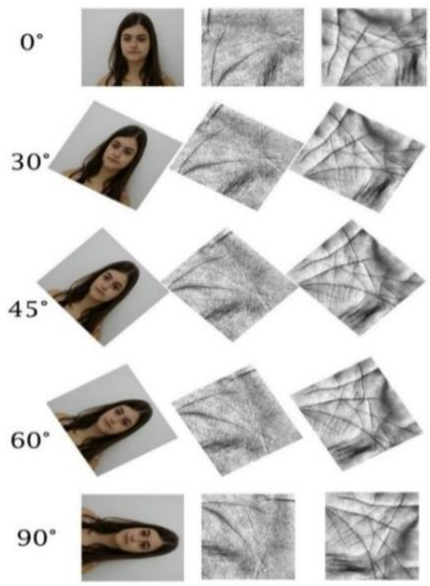


Figure 14. database images using different degrees of rotation

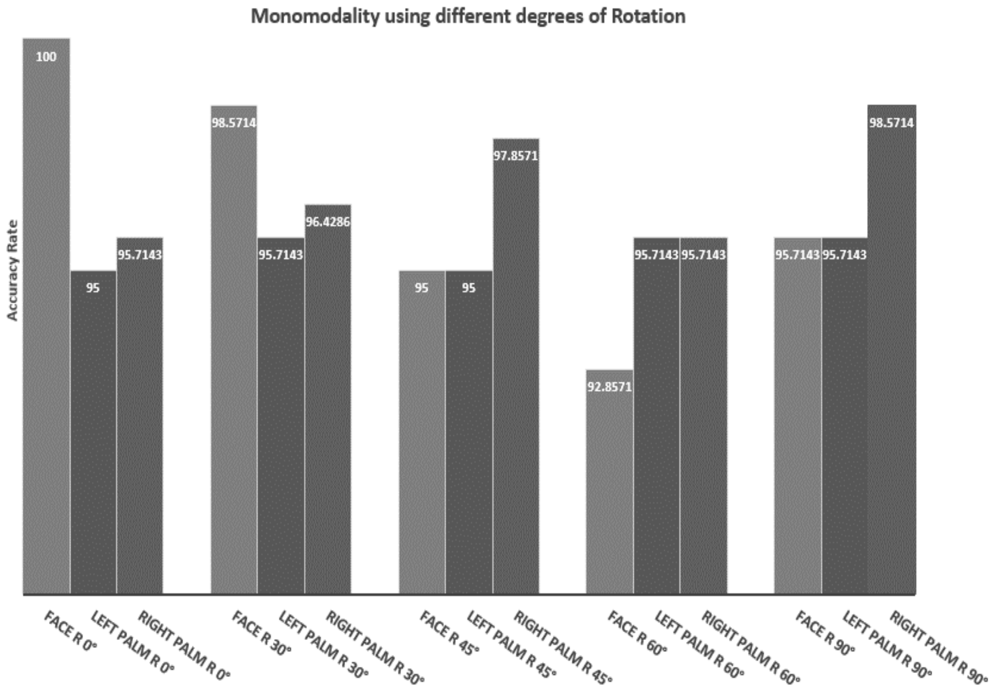


Figure 15. Accuracies rates for Monomodal systems using different degrees of rotation

We notice that the fusion of these traits contributes to decrease the performance of the biometric systems. The results obtained by the monomodal system based on face trait waved between 100% and 92.86% of accuracy rates, as it is shown in the Figure 15. However, the monomodal biometric systems, based on palm maintain performances between 98.57% and 92.86% for the best cases.

Many experiments are done with the rotated data. In general, the obtained results by the fusion of left and right palm prints with different angle of rotation respectively (e.g. 0°+30°, 0°+45%, and so on), achieve 100% of accuracy rates for all the situations. This phenomenon confirms our doubts about the fact that the rotation of these two traits (left and right palm prints) doesn't make sense for recognition and can conduct our biometric systems to over-fitting. And finally, the Figure 16 shows verification accuracy for multi--modal systems using different degrees of rotation.

With similar scenario, we fused the faces (with any rotation) and the left and the right palms. For example, for the two modalities face+(left|right) with a rotation of: 0°+30° and 0°+45°), the models achieve 100% of accuracy in all the situations. This phenomenon can be explained by the fact that the presence of the clean face image (without noise and without rotation) helps to enhance the performance of the multimodal biometric systems as much as possible.

In the Table 4, we present a comparison of our results with other recent works, which is not easy. The used databases, the data quality and the explored algorithms change and vary. However, we notice that our data are augmented and more challenging with adding the noise and the rotation. In addition, the recognition rate obtained with our system based on CNN and KNN is significantly good.

Table 4. Comparison of some recent works, including our system

Modalities	Databases Used	Rate Recognition	Reference
Face-Iris features level	FERET-CASIA v3.0	99.33%	[34]
ECG and Fingerprint decision level fusion feature and level fusion	-CYBHi database and PTB database -LivDet2015 fingerprint database and FVC 2004 database	Less than 100%	[35]
Face-Iris-Fingerprint (features level)	CMU, Multi-PIE, BioCop, and BIOMDATA	99.90%	[36]
Face-Palmprint (features level)	ORL-PolyU and FERET-PolyU	99.7%	[11]
Face-Palmprint (features level)	FERET face and PolyU palm print databases	99.17%	[37]
Face-Palmprint (left and right) Quality: (raw, clean, noisy) without and with: noise and rotation.	FEI face database, IITD Palm print V1	100%	Our System

Figure 16. Fusion left palm print and right palm print with different angle of rotation

5 CONCLUSION

In the present work, multimodal biometric identification systems are proposed using CNN and KNN. The fusion of modalities has proven the strengths of most biometric verification systems when it comes to security matters. The proposed model passed several steps during the design process to determine the best-fit CNN model, as well as the most significant classifier that can be suitable for three types of biometric modalities: face, left palm and right palm prints. The proposed model is then subjected to various types of noise and deformation added to the used data. The results of the conducted experimentations show clearly that the retained system is more resistant to such disturbance in terms of verification performance than any other unimodal biometric system. A de-noising pre-processing of the biometric data seems to be a good initiative to prevent verification performance degradation. The proposed method (CNN and KNN) can be used perfectly for clean and noisy data. Furthermore, future work will emphasize combining other biometrics data such as iris, voice, digital signature and handwriting. A larger-scale application domain such as government biometric data would use huge datasets, so it will be convenient to study the impact of dataset sizes on the performance of such systems. It will be interesting also to investigate other types of machine learning techniques to be associated with these biometric identification systems.

REFERENCES

[1] Y. Wu and Q. Ji, "Facial landmark detection: a literature survey," *International Journal of Computer Vision*, vol. 127, no. 2, pp. 115–142, May 2018, doi: 10.1007/s11263-018-1097-z.

[2] D. S. Trigueros, L. Meng, and M. Hartnett, "Face recognition: from traditional to deep learning methods," Oct. 2018, [Online]. Available: http://arxiv.org/abs/1811.00116.

[3] I. Masi, Y. Wu, T. Hassner, and P. Natarajan, "Deep face recognition: a survey," in *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, Oct. 2018, pp. 471–478, doi: 10.1109/SIBGRAPI.2018.00067.

[4] D. Poux, B. Allaert, J. Mennesson, N. Ihaddadene, I. M. Bilasco, and C. Djeraba, "Facial expressions analysis under occlusions based on specificities of facial motion propagation," *Multimedia Tools and Applications*, vol. 80, no. 15, pp. 22405–22427, Jun.

2021, doi: 10.1007/s11042-020-08993-5.

[5] D. Zhong, X. Du, and K. Zhong, "Decade progress of palmprint recognition: A brief survey," *Neurocomputing*, vol. 328, pp. 16–28, Feb. 2019, doi: 10.1016/j.neucom.2018.03.081.

[6] M. He *et al.*, "Performance evaluation of score level fusion in multimodal biometric systems," *Pattern Recognition*, vol. 43, no. 5, pp. 1789–1800, May 2010, doi: 10.1016/j.patcog.2009.11.018.

[7] M. J. Alhasan, S. M. Abdulhussein, and A. H. K. Khwayyir, "Fingerprint positioning of users devices in long term evolution cellular network using k-nearest neighbour algorithm," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 1, pp. 528–535, Feb. 2021, doi: 10.11591/ijece.v11i1.pp528-535.

[8] E. G. Nihad, E.-N. El Mokhtar, Z. Abdelhamid, and A. A. Mohammed, "Hybrid approach of the fuzzy c-means and the k-nearest neighbors methods during the retrieve phase of dynamic case based reasoning for personalized follow-up of learners in real time," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 6, pp. 4939–4950, Dec. 2019, doi: 10.11591/ijece.v9i6.pp4939-4950.

[9] M. Singh, R. Singh, and A. Ross, "A comprehensive overview of biometric fusion," *Information Fusion*, vol. 52, pp. 187–205, Feb. 2019, doi: 10.1016/j.inffus.2018.12.003.

[10] A. A. Ross and R. Govindarajan, "Feature level fusion of hand and face biometrics," in *Biometric Technology for Human Identification II*, Mar. 2005, vol. 5779, doi: 10.1117/12.606093.

[11] M. I. Ahmad, W. L. Woo, and S. Dlay, "Non-stationary feature fusion of face and palmprint multimodal biometrics," *Neurocomputing*, vol. 177, pp. 49–61, Feb. 2016, doi: 10.1016/j.neucom.2015.11.003.

[12] D. Jagadiswary and D. Saraswady, "Biometric authentication using fused multimodal biometric," *Procedia Computer Science*, vol. 85, pp. 109–116, 2016, doi: 10.1016/j.procs.2016.05.187.

[13] T. A. Alghamdi, "Evaluation of multimodal biometrics at different levels of face and palm print fusion schemes," *Asian Journal of Applied Sciences*, vol. 9, no. 3, pp. 126–130, Jun. 2016, doi: 10.3923/ajaps.2016.126.130.

[14] A. S. Al-Waisy, R. Qahwaji, S. Ipson, and S. Al-Fahdawi, "A multimodal biometrie system for personal identification based on deep learning approaches," in *2017 Seventh International Conference on Emerging Security Technologies (EST)*, Sep. 2017, pp. 163–168, doi: 10.1109/EST.2017.8090417.

[15] N. Alay and H. H. Al-Baity, "A multimodal biometric system for personal verification based on different level fusion of iris and face traits," *Bioscience Biotechnology Research Communications*, vol. 12, no. 3, pp. 565–576, Sep. 2019, doi: 10.21786/bbrc/12.3/3.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Sep. 2014, [Online]. Available: http://arxiv.org/abs/1409.1556.

[17] M. Koziarski and B. Cyganek, "Image recognition with deep neural networks in presence of noise - dealing with and taking advantage of distortions," *Integrated Computer-Aided Engineering*, vol. 24, no. 4, pp. 337–349, Sep. 2017, doi: 10.3233/ICA-170551.

[18] C. Medjahed, F. Mezzoudj, A. Rahmoun, and C. Charrier, "On an empirical study: face recognition using machine learning and deep learning techniques," in *Proceedings of the 10th International Conference on Information Systems and Technologies*, Jun. 2020, pp. 1–9, doi: 10.1145/3447568.3448521.

[19] F. Mezzoudj and A. Benyettou, "An empirical study of statistical language models: n-gram language models vs. neural network language models," *International Journal of Innovative Computing and Applications*, vol. 9, no. 4, 2018, doi: 10.1504/IJICA.2018.10016827.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[21] C. Szegedy *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[23] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, "ImageNet: a large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.

[24] E. Suryawati, R. Sustika, R. S. Yuwana, A. Subekti, and H. F. Pardede, "Deep structured convolutional neural network for tomato diseases detection," in *2018 International Conference on Advanced Computer Science and Information Systems, ICACSIS 2018*, Oct. 2019, pp. 385–390, doi: 10.1109/ICACSIS.2018.8618169.

[25] S. Khan, H. Rahmani, S. A. A. Shah, and M. Bennamoun, "A guide to convolutional neural networks for computer vision," *Synthesis Lectures on Computer Vision*, vol. 8, no. 1, pp. 1–207, Feb. 2018, doi: 10.2200/s00822ed1v01y201712cov015.

[26] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[27] C. M. Bishop, *Pattern recognition and machine learning*, 1st ed. New York: Springer New York, 2006.

[28] M. Z. Al-Dabagh, M. H. Mohammed Alhabib, and F. H. AL-Mukhtar, "Face recognition system based on kernel discriminant analysis, k-nearest neighbor and support vector machine," *International Journal of Research and Engineering*, vol. 5, no. 2, pp. 335–338, Mar. 2018, doi: 10.21276/ijre.2018.5.3.3.

[29] F. Mezzoudj and A. Benyettou, "On the optimization of multiclass support vector machines dedicated to speech recognition," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7664, no. 2, Springer Berlin Heidelberg, 2012, pp. 1–8.

[30] J. A. Lasserre, C. M. Bishop, and T. P. Minka, "Principled hybrids of generative and discriminative models," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, vol. 1, pp. 87–94, doi: 10.1109/CVPR.2006.227.

[31] C. E. Thomaz and G. A. Giraldi, "A new ranking method for principal components analysis and its application to face image analysis," *Image and Vision Computing*, vol. 28, no. 6, pp. 902–913, Jun. 2010, doi: 10.1016/j.imavis.2009.11.005.

[32] A. Kumar and S. Shekhar, "Personal identification using multibiometrics rank-level fusion," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 5, pp. 743–752, Sep. 2011, doi: 10.1109/TSMCC.2010.2089516.

[33] P. Ghuli, S. B. N, and A. G. Rao, "Development of framework for detecting smoking scene in video clips," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 13, no. 1, pp. 22–26, Jan. 2019, doi: 10.11591/ijeecs.v13.i1.pp22-26.

[34] B. Ammour, L. Boubchir, T. Bouden, and M. Ramdani, "Face–iris multimodal biometric identification system," *Electronics*, vol. 9, no. 1, Jan. 2020, doi: 10.3390/electronics9010085.

[35] M. Hammad, Y. Liu, and K. Wang, "Multimodal biometric authentication systems using convolution neural network based on different level fusion of ECG and fingerprint," *IEEE Access*, vol. 7, pp. 26527–26542, 2019, doi: 10.1109/ACCESS.2018.2886573.

[36] S. Soleymani, A. Torfi, J. Dawson, and N. M. Nasrabadi, "Generalized bilinear deep convolutional neural networks for multimodal biometric identification," in *Proceedings - International Conference on Image Processing, ICIP*, Oct. 2018, pp. 763–767, doi: 10.1109/ICIP.2018.8451532.
[37] M. Farmanbar and Ö. Toygar, "Feature selection for the fusion of face and palmprint biometrics," *Signal, Image and Video Processing*, vol. 10, no. 5, pp. 951–958, Jul. 2016, doi: 10.1007/s11760-015-0845-6.

BIOGRAPHIES OF AUTHORS

Chahreddine Medjahed computer science graduate from Mohamed Boudiaf University of Science and Technology Oran Algeria, is currently Ph.D. student in computer science. He is an assistant professor at the Faculty of Computer Sciences of Hassiba Benbouali Chlef, Algeria. His research focuses on Machine Learning and Deep Learning methods and their applications in Image and Video Processing. He can be contacted at email: chahreddine.medjahed@univ-sba.dz.

Abdellatif Rahmoun in the past 20 years; I have been intensively involved in AI research like: neural nets, fuzzy systems, genetic algorithms, genetic programming, hybrid intelligent systems. Presently, working in biometrics, grammar-based genetic programming, strongly typed GP. Recently, I focus on new trends in embedded systems, IoT, smart devices, cloud computing and big data. Presently, I am a full professor at the higher school of computer science (ESI-SBA, Algeria). I am also involved in several international academic societies, as a reviewer or editorial board member in some computer science international journals. You can contact me on my email: a.rahmoun@esi-sba.dz.

Christophe Charrier obtained his Ph.D. in Computer Science from the University Jean Monnet (Saint-Etienne), in 1998. In 2008, he joined the GREYC Lab and the e-payment and biometrics research unit. He researches focuses on Image Quality assessment, Human Vision and Biometrics. He can be contacted at email: christophe.charrier@unicaen.fr.

Freha Mezzoudj received her Ph.D. in 2018 from the University of Science and Technology Oran Mohamed Boudiaf (USTO-MB), Algeria, in the field of Artificiel Intelligence (AI) and Pattern Recognition. I also have received a Magister degree in 2010 from the same University USTO-MB, in the field of AI. Actually, I am an Associate Professor at the department of Computer Sciences of the University Hassiba Benbouali Chlef, Algeria. My subjects of researche are in the fields of Natural Language Processing, Speech Recognition and Machine/Deep Learning. I am the author of more than 10 conference papers and publications. You can contact me on my professional email: f.mezzoudj@univ-chlef.dz.

Fingerprint recognition based on collected images using deep learning technology

Ali Fadhil Yaseen Althabhawee[1], Bashra Kadhim Oleiwi Chabor Alwawi[2]
[1]Directorate General of Education in Holy, Karbala, Iraq
[2]Control and Systems Engineering Department, University of Technology-Iraq, Baghdad, Iraq

Article Info

Article history:

Received Jan 1, 2021
Revised Nov 26, 2021
Accepted Dec 10, 2021

Keywords:

Collected images
Convolutional neural network
Deep learning
Fingerprint
Recognition

ABSTRACT

The fingerprint identification is the most widely used authentication system. The fingerprint uniqueness for each human being provides error-free identification. However, during the scanning process of the fingerprint, the generated image using the fingerprint scanner may differ slightly during each scan. This paper proposes an efficient matching model for fingerprint authentication using deep learning based deep convolutional neural network (CNN or ConvNet). The proposed deep CNN consists of fifteen layers and is classified into two stages. The first stage is preparation stage which includes the fingerprint images collection, augmentation and pre-processing steps, while the second stage is the features extraction and matching stage. Regarding the implantation results, the proposed system provided the best matching for the given fingerprint features. The obtained training accuracy of the proposed model is 100% for training dataset and 100% for validating dataset.

Corresponding Author:

Bashra Kadhim Oleiwi Chabor Alwawi
Control and Systems Engineering Department, University of Technology Iraq
Al-Wehda neighborhood, Box Office 19006, Baghdad, Iraq
Email: 60010@uotechnology.edu.iq

1. INTRODUCTION

Biometrics identification technology are important in security systems of different applications to reduce security threats of military applications, government centers and airports, and criminal applications [1], [2]. Accordingly, the need of safe, reliable, and convenient a identify identification system. Many fingerprint recognition algorithms have been suggested. Park *et al.* [3] proposed a fingerprint recognition system using feature detection algorithm based on scale-invariant feature transform. Cappelli *et al.* [4] presented a new 3D data structure based on minutiae distances and angles named minutiae-cylinder-code. Patil and Suralkar [5] presented fingerprint classification using a neural network. The neural network performed matching step and was successfully used for identifying and classifying the fingerprint utilizing back propagation algorithm. Although many of the previous work has achieved high performance precision, but it involves a lot of pre-processing, which may not be suitable for data sets of different fingerprints under different conditions. recently, efforts have been made for developing recognition models for learning and features extraction automatically [6], [7]. Recently, the emergence of the deep learning technology has great successful achievement in different applications such as image processing, natural language processing and so on [7]. As a results, the fingerprint classification based deep neural network can be also achieved good results. Minaee *et al.* [8] proposed fingerprint recognition model utilizing convolutional neural network (CNN) by tuning trained CNN model on ImageNet and applied the proposed model of the PolyU fingerprint database. Das *et al.* [9] proposed deep learning model based on finger-vein identification and obtained highly accurate and stable performance using different finger-vein images with different quality. Militello *et al.* [10] has presented fingerprint classification systems using deep learning based CNN using two datasets of fingerprint called PolyU and national institute of standards and technology (NIST) and introduced comparisons study with ResNet, AlexNet, and GoogLeNet. Pandya *et al.* [11] has proposed fingerprint classification systems using deep learning based deep CNN using thinning of fingerprint, and Gabor improvement, histogram equalization.

In [12], [13] proposed a new COVID-19 detection model based on histogram techniques and new CNN architecture for improving the images contrast and for COVID-19 detection in chest X-ray image, respectively. Oleiwi and Althabhawee [14] has presented new architecture of CNN for matching the human iris imges in authentication system. In [15], [16] introduced hybrid approach based on deep learning and imag preprocessing for noisy iris dataset. In [17], [18] have proposed sentiment analysis system using deep learning, for Arabic language [17] and for American sign language as static letters (ASL) in hand gesture recognition [18]. In [19], [20] has suggested real time assistive blind model for object detection using deep learning based YOLO algorithm and raspberry model B pi 3. Mahmood and Saud [21] presented monitoring System based detection and classification for moving cars in streaming video. In this paper, the deep learning model based deep CNN is suggested for matching between the samples of presented fingerprint images. The fingerprint database is collected using fingerprint sensor. The fingerprint image augmentation and pre-processing is applied in order to expand and enhance the images. Then, the features extracted using deep CNN and after that, the Softmax performed as a recognizer. The paper is structured as: section 2 introduces the proposed fingerprint recognition system. Section 3 explains the CNN overview and its architecture, and section 4 gives simulation results and discussion. Finally, section 5 presents the conclusion part for the study.

2. THE PROPOSED FINGERPRINT RECOGNITION MODEL

Fingerprint image recognition plays an important role in automated fingerprint identification system, especially in the large dataset. With development of identification system of fingerprint based on big fingerprint database, the accuracy and fast processing time are required. The proposed model is comprised of the following phases.

2.1. Fingerprint dataset acquisition

There are different visual fingerprint patterns that unequally distributed. Thus, the fingerprint types can be divided into five types [2]; the right loop, left loop, the whorl arch and tented arch. In the initialization stage, the fingerprint database of the authorized users was collected using fingerprint sensor named ZKT ZK4500 as a USB fingerprint sensor [22]. The collected dataset includes 1,500 images for 15 authorized users with different ages, each one has 100 samples of his 10 fingers. Some images of the collected dataset are presented in Figure 1, as can be noticed in this figure, there are slightly different colour distributions and different sizes.

Figure 1. Eighteen fingerprint images using fingerprint sensor

2.2. Data augmentation and preprocessing

The fingerprint images augmentation and pre-processing step is important process in order to enlarge and enhance the dataset before feeding the dataset to the network. Essentially it applies random rotations, shifts, flips, crops, and sheers on fingerprint images. The fingerprint images in the dataset are converted from red, green, and blue (RGB) to grayscale images with pixel values ranging from 0 to 255 and resized to a uniform dimension images of size 360×280.

2.3. The matching process

The global and local features of the collected fingerprint images will be extracted using deep CNN. Hence, the representations of the structured feature template data stored in the database, as will be explained in the next section. The matching process is responsible for comparing the stored image and inputted image in order to achieve a verdict match or non-match. Hence, the entered fingerprint image scanned using fingerprint sensor and the feature extraction implemented through the same process in the initialization stage, as illustrated in Figure 2.

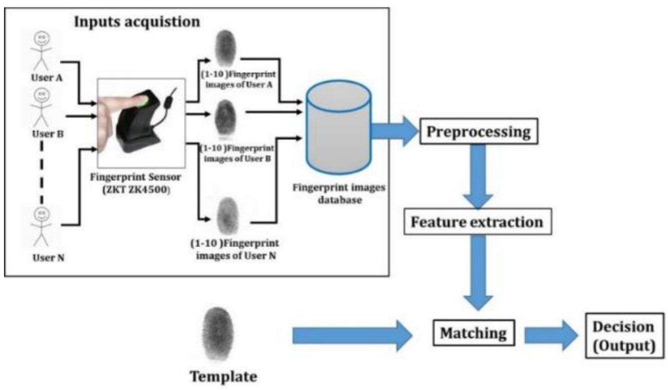


Figure 2. General block diagram of the proposed recognition system

3. OVERVIEW OF CNN

The deep learning field represents a subfield of machine learning concerning with algorithms that inspired by human brain structure and function which is named artificial neural networks. Deep-learning methods such as CNN consists of a multi convolutional layers with a fully connected layer, based a robust method of feature extraction and classification [9], [23], [24]. The CNN layers will be explained as:

– Convolutional layers: represent a several 2-D convolutional layers started from the input maps x_m^l, where l and m represent indexes of the level and map respectively. The filters based on the kernels $w_{n,m}^l$, where *n* is the index of the filter. The nth is the output map y_n^l of layer l which can be calculated as [9], [25]:

$$y_n^l = \sum_{m}^{M^{l-1}} w_{n,m}^l * x_m^l + b_n^l \tag{1}$$

Where M^{l-1} denotes input maps number, * is convolution, and b_n^l, represents bias of *nth* map output in the *lth* level.

– The rectified linear unit (ReLU) layer: represents activation layer as nonlinear layer which is used directly after the convolutional layer mentioned by (1). This layer allows model to train faster with high accuracy. By applying activation function in (2).

$$Softmax[f(x) = max(0, x)] \tag{2}$$

which will be increased as non-linearity.

– Pooling layer: This layer minimizes the spatial size, by reducing the required parameters number for describing the network, which led to reduce the computational process that are required for training the model.

– Fully connected layer: fully connected of all neurons as in ordinary neural systems. It represents the last part of the network, for yielding the probabilities of the class. The linear combination O_n is expressed in (3), where x_m is the mth map input in the output layer [9], [26]. The high level features will be extracted in this layer by calculating probabilities of the available classes. The probability distribution of the input data over C different classes will be predicted by using the Softmax function, as defined in (4):

$$O_n = \sum_{m=1}^{M} (w_{n,m} * x_m + b_n) \quad (3)$$

$$p_u = \frac{\exp(O_u)}{\sum_{n=1}^{C} \exp(O_n)} \quad (4)$$

3.1. Designed CNN architecture

The structure of the designed CNN for performing fingerprint recognition system consists mainly of convolutional layer, pooling layer, fully connected layer and the activation functions as illustrated in Figure 3. The defined 2 D CNN strucure has fifteen Layers: image input as gray image with size 360, 280, b1, then 3 convolution layers for extracting the features based on different sizes, strides and padding, followed by 3 RelU layers as activation functions, after that 3-max pooling layeras for reducing the parameters, fully connected layers, softmax layer and classification layer as introduced in Figure 4. Figure 5 represents the options set chosen for training the proposed CNN.

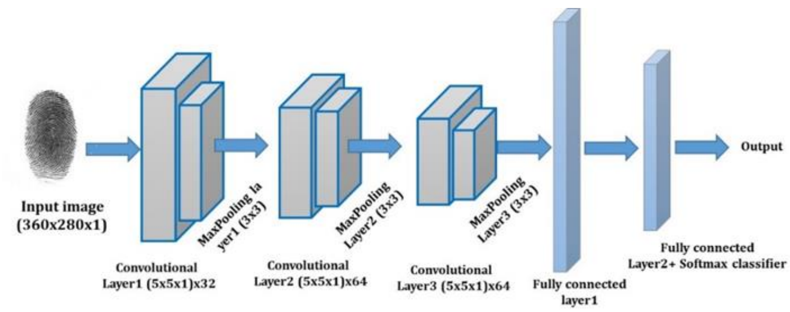


Figure 3. Employed CNN architecture

```
layers =

  15x1 Layer array with layers:

     1   ''   Image Input             360x280x1 images with 'zerocenter' normalization
     2   ''   Convolution             32 5x5 convolutions with stride [1  1] and padding [2  2  2  2]
     3   ''   ReLU                    ReLU
     4   ''   Max Pooling             3x3 max pooling with stride [2  2] and padding [0  0  0  0]
     5   ''   Convolution             32 5x5 convolutions with stride [1  1] and padding [2  2  2  2]
     6   ''   ReLU                    ReLU
     7   ''   Max Pooling             3x3 max pooling with stride [2  2] and padding [0  0  0  0]
     8   ''   Convolution             64 5x5 convolutions with stride [1  1] and padding [2  2  2  2]
     9   ''   ReLU                    ReLU
    10   ''   Max Pooling             3x3 max pooling with stride [2  2] and padding [0  0  0  0]
    11   ''   Fully Connected         64 fully connected layer
    12   ''   ReLU                    ReLU
    13   ''   Fully Connected         3 fully connected layer
    14   ''   Softmax                 softmax
    15   ''   Classification Output   crossentropyex
```

Figure 4. The defined CNN fifteen layers

3.2. Software part of the CNN

The basic steps of the software part of the fingerprint identification system using CNN based multi layers are presented in flowchart in Figure 6. In the first step of this flowchart, the inputted image was recorded and read, then converted it from a color image to a gray image, and then CNN is applied for the purpose of detecting and classifing the inputted fingerprint if it matches the stored fingerprints or not. In the last stage, the detected image is classified into which class belongs of the stored fingerprints with rating ratio for each fingerprint stored in the dataset.

```
options =

  TrainingOptionsSGDM with properties:

                    Momentum: 0.9000
            InitialLearnRate: 1.0000e-04
    LearnRateScheduleSettings: [1×1 struct]
            L2Regularization: 1.0000e-04
                   MaxEpochs: 300
               MiniBatchSize: 128
                     Verbose: 0
            VerboseFrequency: 50
              ValidationData: [1×1 matlab.io.datastore.ImageDatastore]
         ValidationFrequency: 30
          ValidationPatience: 5
                     Shuffle: 'once'
              CheckpointPath: ''
        ExecutionEnvironment: 'cpu'
                  WorkerLoad: []
                   OutputFcn: []
                       Plots: 'training-progress'
              SequenceLength: 'longest'
        SequencePaddingValue: 0
```

Figure 5. The options set chosen for training the CNN network

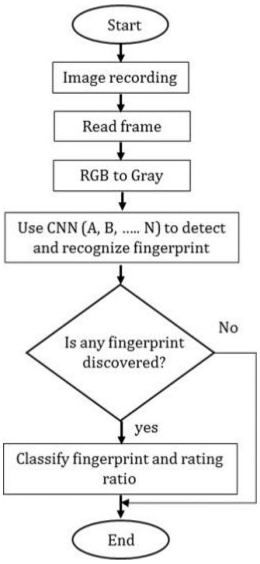


Figure 6. Flowchart of the fingerprint identification system using CNN

4. RESULTS AND DISCUSSION

The implementatin of the introduced model required two parts, software and hardware. The software part using MATLAB version 2020b [25], [27], and the hardware part using a PC assembled specially for the work with core i5 CPU composed of four 3.2 GHz working cores, a 10 GB RAM, and an advanced GPU of 6.1 computation capability measure. The proposed model are implemented in parallel computing fashion in which the four CPU cores and the GPU cooperate together.

In the initialization stage, the fingerprint images of the authorized users will be collected by fingerprint sensor. The features of the collected fingerprint image will be extracted, and then the comparison step in the matching stage. The template or inputted image will be compared with the stored dataset in order to achieve the matching outcome process. Hence, the query fingerprint image is scanned using fingerprint sensor. The feature extraction of the query fingerprint image implement using the same procedures in the initialization stage. The simulation has been carried out to evaluate the performance of the proposed system based on special fingerprint database. The fingerprint images have been acquired and quantized according to 360×280 and 256 gray. Figure 7 includes of patterns of the database that has been used for training process. While Figure 8 represents the performance of the proposed system, Figure 8(a). Training on single GPU, Figure 8(b). Training progress. Finally, the Fingerprint recognition has been represented in Figure 9. The obtained results indicated the success of the proposed model in achieving its intended in classification and identification of the fingerprint. Hence, the proposed system managed to achieve complex computations in relatively short times.

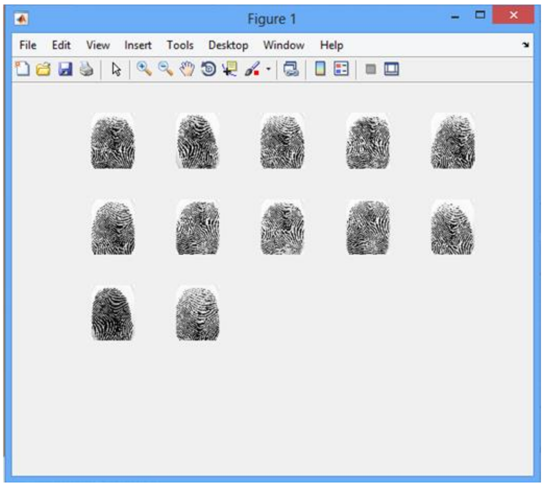
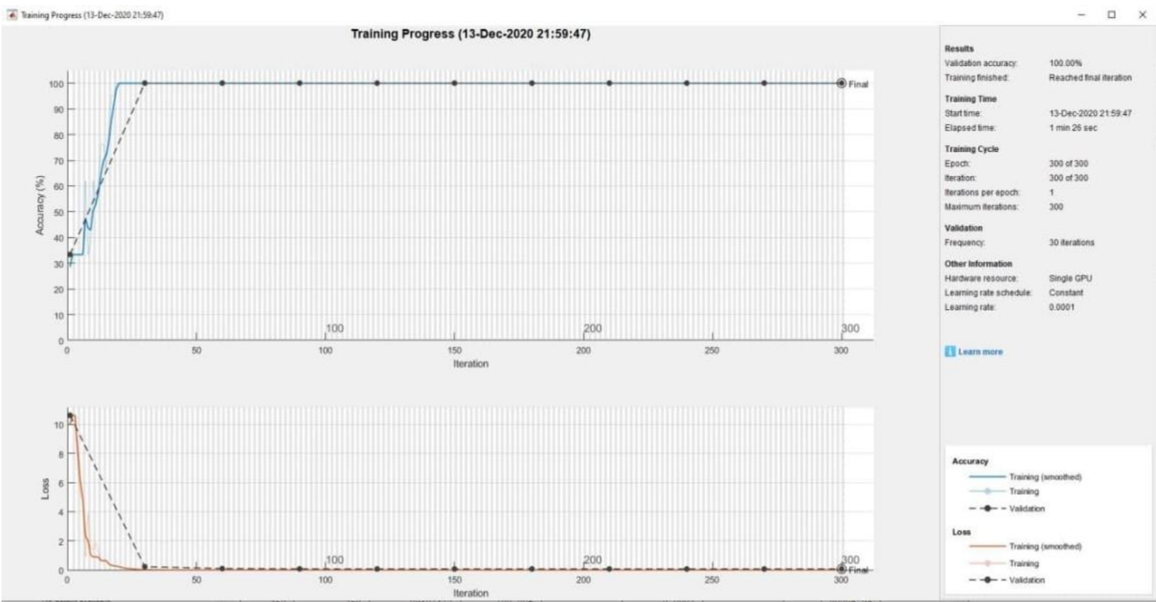


Figure 7. Patterns of the database that has been used for training process

Training on single GPU.
Initializing input data normalization.

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Validation Accuracy	Mini-batch Loss	Validation Loss	Base Learning Rate
1	1	00:00:01	28.57%	33.33%	10.0100	10.6283	1.0000e-04
30	30	00:00:10	100.00%	100.00%	0.0136	0.2199	1.0000e-04
50	50	00:00:15	100.00%		0.0009		1.0000e-04
60	60	00:00:18	100.00%	100.00%	0.0006	0.0929	1.0000e-04
90	90	00:00:27	100.00%	100.00%	0.0004	0.0709	1.0000e-04
100	100	00:00:30	100.00%		0.0004		1.0000e-04
120	120	00:00:36	100.00%	100.00%	0.0003	0.0674	1.0000e-04
150	150	00:00:44	100.00%	100.00%	0.0003	0.0669	1.0000e-04
180	180	00:00:52	100.00%	100.00%	0.0003	0.0664	1.0000e-04
200	200	00:00:58	100.00%		0.0002		1.0000e-04
210	210	00:01:01	100.00%	100.00%	0.0002	0.0659	1.0000e-04
240	240	00:01:09	100.00%	100.00%	0.0002	0.0656	1.0000e-04
250	250	00:01:12	100.00%		0.0002		1.0000e-04
270	270	00:01:18	100.00%	100.00%	0.0002	0.0654	1.0000e-04
300	300	00:01:26	100.00%	100.00%	0.0002	0.0651	1.0000e-04

(a)



(b)

Figure 8. The performance of the proposed system for (a) training on single GPU and (b) training progress

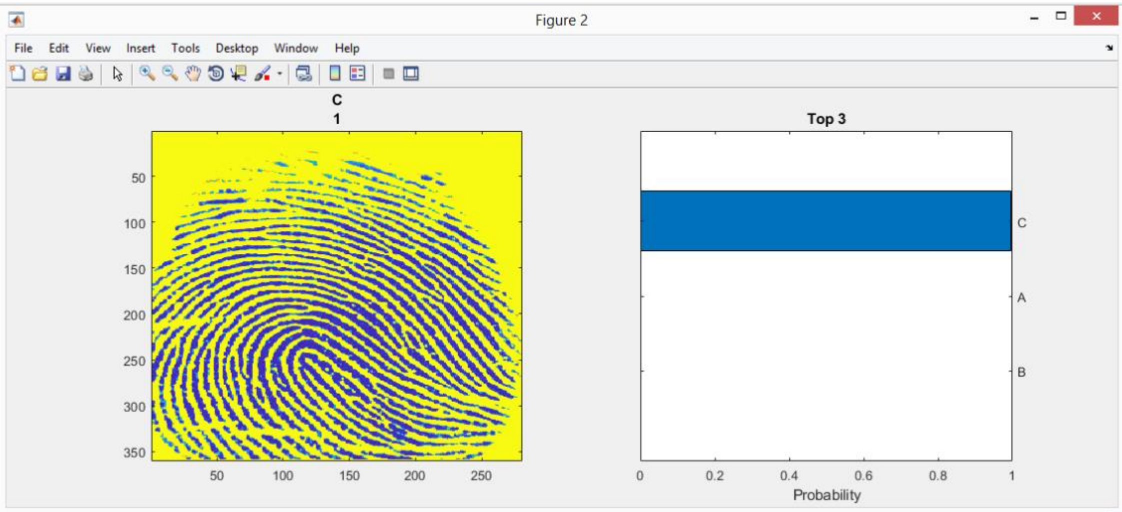


Figure 9. Fingerprint classification

5. CONCLUSION

In this paper, a deep learning framework for fingerprint recognition using CNN to learn the representation of the features and perform the recognition. The proposed model has been trained on a large-scale fingerprint recognition dataset. The successful simulation results have been indicated that the proposed system can be applied as a reliable system for developing artificial identification and can be adapted to any image contents of different characteristics. The accuracy of the proposed system according to training model was 100% for training dataset and 100% for validation dataset. Therefore, the proposed system is robust, reliable, efficiency. In future research, CNN structure optimization will be study to speed up the learning speed for fast matching and to improve the recognition performance of images with noise, to apply multi modal biometric recognition and implement in real time using Raspberry pi and fingerprint sensor.

REFERENCES

[1] R. Wang, C. Han, Y. Wu, and T. Guo, "Fingerprint classification based on depth neural network," 2014, [Online]. Available: http://arxiv.org/abs/1409.5188.

[2] D. Peralta, I. Triguero, S. García, Y. Saeys, J. M. Benitez, and F. Herrera, "On the use of convolutional neural networks for robust classification of multiple fingerprint captures," *International Journal of Intelligent Systems*, vol. 33, no. 1, pp. 213–230, 2018, doi: 10.1002/int.21948.

[3] U. Park, S. Pankanti, and A. K. Jain, "Fingerprint verification using SIFT features," in *Biometric Technology for Human Identification V*, 2008, vol. 6944, doi: 10.1117/12.778804.

[4] R. Cappelli, M. Ferrara, and D. Maltoni, "Minutia cylinder-code: a new representation and matching technique for fingerprint recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2128–2141, 2010, doi: 10.1109/TPAMI.2010.52.

[5] S. R. Patil and S. R. Suralkar, "Fingerprint classification using artificial neural network," *International Journal of Scientific and Advanced Engineering*, vol. 2, no. 12, pp. 1–4, 2012.

[6] W. Yang, S. Wang, J. Hu, G. Zheng, and C. Valli, "Security and accuracy of fingerprint-based biometrics: a review," *Symmetry*, vol. 11, no. 2, Jan. 2019, doi: 10.3390/sym11020141.

[7] S. Minaee, A. Abdolrashidi, H. Su, M. Bennamoun, and D. Zhang, "Biometrics recognition using deep learning: a survey," Nov. 2019, [Online]. Available: http://arxiv.org/abs/1912.00271.

[8] S. Minaee, E. Azimi, and A. Abdolrashidi, "FingerNet: pushing the limits of fingerprint recognition using convolutional neural network," Jul. 2019, [Online]. Available: http://arxiv.org/abs/1907.12956.

[9] R. Das, E. Piciucco, E. Maiorana, and P. Campisi, "Convolutional neural network for finger-vein-based biometric identification," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 2, pp. 360–373, 2018, doi: 10.1109/TIFS.2018.2850320.

[10] C. Militello, L. Rundo, S. Vitabile, and V. Conti, "Fingerprint classification based on deep learning approaches: experimental findings and comparisons," *Symmetry*, vol. 13, no. 5, p. 750, Apr. 2021, doi: 10.3390/sym13050750.

[11] B. Pandya, G. Cosma, A. A. Alani, A. Taherkhani, V. Bharadi, and T. M. McGinnity, "Fingerprint classification using a deep convolutional neural network," in *2018 4th International Conference on Information Management, ICIM 2018*, 2018, pp. 86–91, doi: 10.1109/INFOMAN.2018.8392815.

[12] B. K. O. C. Alwawi and L. H. Abood, "Convolution neural network and histogram equalization for COVID-19 diagnosis system," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 24, no. 1, pp. 420–427, Oct. 2021, doi: 10.11591/ijeecs.v24.i1.pp420-427.

[13] B. K. O. and M. I. AL Tameemi, "Visual-infrared video monitoring system for moving object detection, classification and tracking using deep learning technique," *8th Engineering and 2nd International Conference for College of Engineering*, 2021.

[14] B. K. Oleiwi and A. F. Althabhawee, "Towards more accurate and efficient human iris recognition model using deep learning technology," *TELKOMNIKA Telecommunication, Computing, Electronics and Control*, 2021.

[15] H. J. A. A. T. H. A. M. Hasan, "Segmentation approach for a noisy iris images sased on hybrid techniques," *Engineering and*

Technology Journal, vol. 38, pp. 1684–1691, 2020, doi: 10.30684/etj.v38i11A.450.
[16] A. T. Hashim and D. A. Noori, "An approach of noisy color iris segmentation based on hybrid image processing techniques," in *Proceedings - 2016 International Conference on Cyberworlds, CW 2016*, 2016, pp. 183–188, doi: 10.1109/CW.2016.39.
[17] A. Nasser and H. Sever, "A concept-based sentiment analysis approach for Arabic," *International Arab Journal of Information Technology*, vol. 17, no. 5, pp. 778–788, 2020, doi: 10.34028/iajit/17/5/11.
[18] A. Abdulhussein and F. Raheem, "Hand gesture recognition of static letters american sign language (ASL) using deep learning," *Engineering and Technology Journal*, vol. 38, no. 6, pp. 926–937, 2020, doi: 10.30684/etj.v38i6a.533.
[19] B. K. O. and M. R. Kadhim, "Real time embedded system for object detection using deep learning," *The 2nd International Conference on Electromechanical Engineering & its Applications, ICEMEA*, 2021.
[20] M. R. K. and B. K. Oleiwi, "Blind assistive system based on real time object recognition using machine learning," *Engineering and Technology Journal, an accepted paper*, 2021.
[21] S. Mahmood and L. Saud, "An efficient approach for detecting and classifying moving vehicles in a video based monitoring system," *Engineering and Technology Journal*, vol. 38, no. 6, pp. 832–845, 2020, doi: 10.30684/etj.v38i6a.438.
[22] Zkteco.systems, "Zk4500." https://zkteco.systems/en/product/zk4500/.
[23] J. Murphy, "An overview of convolutional neural network architectures for deep learning," *Microway*, 2016, doi: 10.20944/preprints201811.0546.v1.
[24] M. Thoma, "Analysis and optimization of convolutional neural network architectures," Jul. 2017, [Online]. Available: http://arxiv.org/abs/1707.09725.
[25] Mathworks.com "Convolutional neural network," *Mathworks*, 2021 [Online]. Available:. https://www.mathworks.com/discovery/convolutional-neural-network-matlab.html.
[26] R. Vidal, J. Bruna, R. Giryes, and S. Soatto, "Mathematics of deep learning," 2017, [Online]. Available: http://arxiv.org/abs/1712.04741.
[27] Mathworks.com "Deep learning toolbox," *Mathworks. com* ,2021 [Online]. Available. https://www.mathworks.com/products/deep-learning.html.

BIOGRAPHIES OF AUTHORS

Ali Fadhil Yaseen Althabhawee born in Karbala-Iraq, completed B.Sc. degree in Computer Engineering at Alhussain University college, Karbala-Iraq. He finished his M.Sc.degree in Computer Engineering at University of Technology, Bagdad-Iraq. His research interests are artificial intelligence systems and deep learning and machine learning and programming languages. He is currently working as computer engineer at the Ministry of Education Directorate in Iraq. He can be contacted at email: 61124@student.uotechnology.edu.iq

Bashra Kadhim Oleiwi Chabor Alwawi completed a Master degree in Mechatronics Engineering/Control and Systems Engineering Department at University of Technology (UOT) Bagdad-Iraq. She finished Ph.D. student at Control Engineering Department (RST), Siegen University, Germany. Her research interests are mobile robot, path planning, multi objective optimization and artificial intelligence systems and deep learning and machine learning. She is currently working as a faculty member in control and systems engineering department at UOT. She can be contacted at email: 60010@uotechnology.edu.iq

IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 11, No. 1, March 2022, pp. 89~101
ISSN: 2252-8938, DOI: 10.11591/ijai.v11.i1.pp89-101

Bi-directional long short term memory using recurrent neural network for biological entity recognition

Rashmi Siddalingappa, Kanagaraj Sekar
Department of Computational and Data Sciences, Indian Institute of Science, Bangalore, India

Article Info

Article history:

Received Jun 20, 2021
Revised Dec 15, 2021
Accepted Dec 27, 2021

Keywords:

1-hot vector representation
Bi-directional recurrent neural network
Electronic medical records
GloVe
Long-short-term-memory
Named-entity recognition
Skip-gram model

ABSTRACT

Biomedical named entity recognition (NER) aims at identifying medical entities from unstructured data. A quintessential task in the supervision of biological databases is handling biomedical terms such as cancer type, DeoxyriboNucleic and RiboNucleic Acid, gene and protein name, and others. However, due to the massive size of online medical repositories, data processing becomes a challenge for a gazetteer without proper annotation. The traditional NER systems depend on feature engineering that is tedious and time-consuming. The research study presents a new model for Bio-NER using recurrent neural network. Unlike existing approaches, the proposed method uses bidirectional traversing with GloVe vector modelling performed at character and word levels. Bio-NER is performed in three stages; firstly, the relevant medical entities in electronic medical records from PubMed were extracted using the skip-gram model. Secondly, a vector representation for each word is created through the 1-hot method. Thirdly, the weights of the recurrent neural network (RNN) layers are adjusted using backward propagation. Finally, the long-short-term memory cells store the previously encountered medical entity to tackle context-dependency. The accuracy and F-score are calculated for each medical entity type. The MacroR, MacroP, and MacroF are equal to 0.86, 0.88, and 0.87. The overall accuracy achieved was 94%.

Corresponding Author:

Rashmi Siddalingappa
Department of Computational and Data Sciences, Indian Institute of Science
Bangalore-560012, Karnataka, India
Email: drrashmis64@gmail.com

1. INTRODUCTION

The electronic medical records (EMRs) are a vast repository of data maintained by healthcare professionals over a long period containing vital information about a patient in structured and unstructured formats. They include patients' medical conditions, treatments, progress notes, physical condition, prognostic and diagnostic procedures, past medications, immunization, lab reports, discharge indications, and persistence of any other medical problems [1]. Artificial intelligence (AI), particularly natural language processing (NLP), helps discover various associations among these parameters, providing clinicians with opportunities to improve treatment outcomes and administer systematic medical delivery [2]. Bio-named entity recognition (NER) is a process of identifying, classifying, and tagging medical entities for a given context [3]. The main challenges of Bio-NER are: i) tagging entities for the same class (protein/ deoxyribonucleic acid (DNA) or cell type/cell line) does not confer to a standard naming convention; ii) there are a plethora of abbreviations used for different compounds for both long and short words; iii) use of special symbols such as hyphen, colon, and greek letters could create different annotations leading to defiance for a NER task; and iv) the technical terms in online repositories are increasing, the tenacity of disambiguating medical terms becomes cumbersome for a gazetteer. The authors in [4] describe different methods to build an efficient Bio-NER system. The work concentrated on marking tags for bio-entities (DNA and protein terms) using conditional random fields (CRFs) and recurrent neural network (RNN). However, the baseline accuracy obtained on BioNLP 2004 corpus was 70.09% since the RNN does not retain the previous information. The traditional approaches to handle Bio-NER are: i) dictionary/corpus-based: here, a corpus contains entities and their associated tags. The target word is searched, and the corresponding label is retrieved from the corpus. Though this approach is simple, the efficiency is dependent on the type and size of the corpus. If a target word is not present, the system will not tag the entity. For instance, if the dictionary contains a protein name as “NF-Kaapa B” and the input context has “NF KappaB,” then the system will not recognize this variation. Further, the approach also suffers from false recognition and low recall values, especially for small words and spelling variations [5]; ii) rule-based: hand-generated rules are created by a gazetteer. Though the accuracy is increased, manually creating rules is cumbersome and depends on expert knowledge and domain [6]; and iii) machine learning-based: the statistical models in machine learning (ML) recognize entities through the feature representation. The primary steps are training and testing; training, where annotations are marked based on an annotated document and later store the model; next, the annotations for the raw document are scored based on the model saved. This approach is the best compared to other methods. Here, the model can recognize new tags even when they are not present in the corpus [7]. Recent deep learning methods with traditional ML models are amalgamated to achieve good results [8]. Deep learning mimics human brain functions to help process data by creating functional patterns to support decision-making. More interestingly, deep learning techniques outperform other ML-based approaches such as support vector machines (SVM) [9], hidden markov model (HMM) [10], maximum entropy likelihood model conditional random fields [11].

Deep learning has recently led its way in Bio-NER, too, and researchers have achieved excellent results [12], [13]. Bio-NER to identify genes and proteins using RNN is proposed by Li *et al.* [14]. The information from the last node is considered to make new entity predictions. An extra class-based input layer was created using a brown clustering algorithm. The features are extracted using word-embedding, and the work is demonstrated on BioCreative II genetic mutation (GM) and obtained an accuracy of 81.06% on f-score. Though the extended-RNN framework attained a better accuracy, perhaps an additional feature layer could incur extra processing time. Ali *et al.* [15] have used sequence labelling and memory component of RNN to mark Arabic text labels. A pre-trained embedding using the LSTM network is used for training the corpus. The model is evaluated on Arabic-NER Corp achieving an f-score of 88.01%. Cho and Lee [16] focused on marking bio-medical entities using bi-direction long short term memory (BiLSTM). The critical associations between the adjacent labels were drawn using condition random fields (CRFs). The computational time of the contextual LSTM (CLSTM) network was computed, and it was compared with other models (bidirectional encoder representations from transformers (BERT) and bidirectional recurrent neural network (BiRNN)). It was found that CLSTM had a faster training time as compared to different character-level embeddings. However, it took 20% longer training time than the BiRNN models. The results are evaluated on three corpora; the National Center for BioTechnology, Gene Mutation, and Chemical Disease-Related Corpus, and the accuracy of 85% was recorded. Lyu *et al.* [17] proposed bioNER using CRFs on word-character representation. The CRF layer encodes the context information of a given sentence in two directions, forward and backward. The model was tested against two corpora, bio creative II GM and joint workshop on NLP (2004), and achieved an accuracy of 86.55% and 73.79%, respectively. The approach performed better than other models; however, contextual and external knowledge information was not considered. Bio-NER for Chinese texts was designed by Li *et al.* [18]. The orthographic and lexicon-semantic features were derived from the given context using the word (W) and character (C) embedding. The part-of-speech (POS) tags (P) of previous information are used to improve the overall performance and obtained an accuracy of 90%. However, the domain-specific features related to diagnosis or medications were not included in this WCP-RNN based research study. Chowdhury *et al.* have proposed a multitask Bi-RNN model for Chinese EMRs [19]. The work was divided into the shared and task-specific layers. Each word was represented using word and character embedding. The context information was extracted using Bi-RNN. In the next layer, POS tagging was marked to separate the POS tags from the given context, and in the next step, NER was performed for identifying the entities. This approach requires high training time as it contains two extra task-specific layers for POS tagging. NER implementation for national center for biotechnology information (NCBI)-disease and JNLPBA corpus (Joint Workshop on Natural Language Processing in Biomedicine and its Applications) have been discussed by authors in [20] using LSTM and convolutional neural network (CNN) models. The proposed model lacked the knowledge transferring approach. The accuracy was equal to 74.4% and 86.0%, respectively, for these binary datasets. Further, the authors in [21] have explained different efficient methods ruling in the industry for Bio-NER and relation detection (RD) to learn the interaction between protein, drug, diseases, and genes. In the present research paper, the authors have tried to overcome the shortcomings of the existing literature in the following ways: i) by far, the global vector representation (GloVe) representation model is not used to address BioNER. The features are extracted at both character and word levels using GloVe embeddings. Thus, the accuracy was increased, unlike other state-of-art systems; ii) no additional layer was required for training features in the hidden layer; iii) the POS labels for all the entities are not marked, except for bio-medical phrases; thereby, the execution time gradually decreased; and iv) the LSTM layers treat each entity class as a different model; thus, the disease name is not confused with the drug or gene name. Consequently, each layer becomes a master in the training time and a candidate during testing time.

The objectives of the present work are: i) recognize the medical entities and the association between other vital terms like genes/proteins causing cancer or mutations indicating the up or down-regulation in cancer; ii) disease identification; iii) adopting deep learning strategies for perceiving the relevant features to understand the critical role played by cancer entities; iv) to automatically extract meaningful bio-entities found in unstructured data with biological relevance; v) to learn and annotate various biological terms from the biomedical repositories such as PubMed; and vi) study the efficiency of the model through evaluation metrics. The paper is structured as indicated here: section 2 talks about research methods such as different stages of the Bi-RNN and LSTM frameworks. Then, the experimental and implementation results are described in section 3. Finally, the study concludes with section 4.

2. RESEARCH METHODOLOGY

The present “problem-solution” research paper explains the problems faced by the existing bio-ner systems and finds suitable solutions using the advanced deep neural network model. In the pursuit of this, the following details are discussed in this section: i) methods adopted: RNN for learning features, GloVe embeddings at character and word level to extract essential features, and LSTM to retain the critical features and forget those features that are no more essential for learning tasks; ii) the bi-directional RNN and LSTM algorithms and pseudo codes; and iii) evaluation metrics: F-score components

2.1. Recurrent neural network and components

The RNN model uses sequential information to process text. In a traditional neural network, an input at each layer is self-reliant and does not depend on other layers’ inputs. However, this is unlikely in a large number of prediction tasks. For instance, knowledge about the previous word is essential in predicting the next word of the sequence. Thus, in RNNs, the previous output is treated as inputs for the present state [22]. Besides, RNN also has a memory element to capture the computations performed during the pre-trained embedding task. A typical RNN model is shown in Figure 1. The layers of an RNN model is dependent on the number of input words in a given sentence ‘S’ such that each word is represented by ‘W’The terms used are described below.

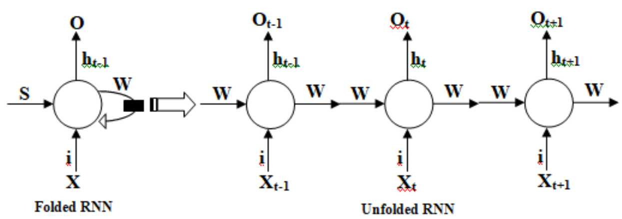


Figure 1. Basic Structure of an RNN model with two variations; folded and unfolded model

- x<t>: the 1-hot vector representation of a given word at time ‘t’
- h<t>: the hidden state at ‘t’ that contains the output of the previous state at time ‘t-1’ and the input of the current state at the time ‘t’. It is usually calculated as h<t>=tanh(cx<t>+ ph<t-1>), where ‘c’ indicates the current input at x<t> time and ‘p’ indicates the previous output at hidden state h<t-1>. Tanh is a non-linear function for activating input and hidden states.
- o<t>: output obtained at each layer in the time step <t>. If the bio-entity tagging is expected, then output at <t> step would be series of vector probabilities in the chosen corpus ‘C’ at the training stage. Softmax, expressed as o<t>=softmax[Ch<t>], is the activation function used for normalizing output value. Consider the following example for bio-ner task: “chronic myeloid leukemia (CML) is characterized by the presence of a breakpoint cluster-abelson kinase (BCR-ABL) fusion gene. Being an inhibitor of BCR-ABL, Imatinib rapidly and dramatically modified CML treatment. Nilotinib and

Dasatinib also have superior efficacy to Imatinib to diagnose CML. Gastrointestinal stromal tumors (GISTs) are defined by C-KIT expression (CD117) in tumor cells. Imatinib was found effective in patients carrying KIT mutations. Thus, the tyrosine kinase (TK) inhibitor, Imatinib has revolutionized the therapy of malignancies that are addicted to one of its target kinases, C-ABL, C-KIT and platelet-derived growth factor receptor (PDGFR)" [23]. * *ABL: Aberlson murine leukemia, C-KIT: cluster of differentiation 117- tyrosine-protein kinase*

In the above example, the dependency between each term is evident. Initially, CML is tagged as a cancer type. The same word CML appears after a long gap, enforcing a long-term dependency. RNN model operates smoothly only for trigram context dependency. RNN loops the input repeatedly, modifying the input weights/gradients. These gradients refer to the values used to update the network weights at each input layer. A substantial change in these input values leads to error gradients, which results in a poor network. An unstable network constitutes two main challenges; firstly, the gradient values can grow exponentially, at times greater than 1, because they multiply at each layer with the hidden values, leading to overflow or Nan values. Secondly, if the values are smaller than 1, they may quickly vanish due to recurrently operating on a smaller value. A low gradient value does not help much in the learning process. Thus, the LSTM technique is used to handle this problem. LSTM retains only the relevant information and forgets irrelevant data. For instance, LSTM considers words such as "being, was, an, have, are, dramatically, rapidly, and so on" as irrelevant. In addition to this, the forward pass of a conventional RNN model considers the data processing in the single, sequential direction and fails to look ahead for the context-dependency. On the other hand, Bi-RNN "looks ahead" for any dependencies between the key terms. For example, i) Teddy Douglas was the president during the drought and ii) she was excited to know that Teddy bear dolls were on sale at the sixth main avenue. In these examples, the word 'Teddy' indicates the person's name and a toy. In a unidirectional approach, the system would tag 'Teddy' as a toy. However, in a BiRNN system, the model looks forward and learns the context information and accordingly marks the tag for 'Teddy' as the person's name in example 1 and a toy in example 2. A loss function is used in the backward pass to quantify the wrongness of the model. A loss is measured as a square difference between the answer obtained and the expected correct output. Suppose a BiRNN model calculated output value as 0.6, and the result expected is 1; the loss is calculated as $[0.6-1]^2=0.16$. This value suggests that the weights have to be tweaked to minimize the value of the loss function. Thus, a backward pass is also referred to as gradient descent with the objective to descent (minimize) the gradients (weights) [24]. These steps are performed for an ideal number of times called the epoch, from which the network gets better every time.

2.2. Extracting features at word and character level using GloVe model

Tokenization is performed in step-1. Each token then serves as input for word embedding, Figure 2. The figure indicates the following terms: $<F_i>$: forward-pass features, $<B_i>$: backward-pass features and $<W_i>$: word-entity concatenation, d X T: input gradient at time-step. Here, a word is represented in a vector form, i.e., a unique number, based on the corpus. For instance, the word imatinib appears in 5536th place in the corpus. Therefore, the 1-hot representation is Imatinib=$O_{5536.}$ Word-vectors cluster similar words and dissimilar words are repelled. Consider the subset of the example; imatinib treats CML. Nilotinib also treats leukemia. Oxaliplatin is used against Colorectal Cancer. The word and the dimensional vector are indicated in Table 1. By looking at the high numbers in the Table 1, we can say that 'treat' is closely associated with Imatinib, nilotinib, and leukemia, indicating that these terms are closely used for treating some 'X' (X is unknown, it could be a disease). Values of CML, cancer, colorectal, and leukemia are higher in the second row, extrapolating the disease names. The dimensional vector represents how closely the terms (row x column) are associated.

The GloVe model is used as a word-embedding task [25]. Unlike the traditional word2vec method, the GloVe model looks for global co-occurrence between words and later mark vectors. The GloVe model forecasts the input texts to their corresponding vector values. Here, certain dimensionality constraints are defined randomly. The dimensional vectors are; 'treat,' 'cancer,' drug names, and so on. These keywords are placed in a columnar fashion. For each dimensional vector, the value of a keyword is compared using a random function. For instance, Imatinib is used to treat cancer. Therefore, the number 1 indicates a valid substitution. However, CML is a cancer and is not a 'treat' entity. Therefore, a negative value is assigned. The 2d space representation of this analogy is indicated in Figure 3. Thick lines indicate a solid word-pair association, and dotted lines indicate a weak connection. With this, we can conclude that, for CML, imatinib is used, and similarly, for colorectal cancer, oxaliplatinis used.

Internally, $E_{Imatinib}$-E_{CML}~$E_{Oxaliplatin}$–$E_{?}$. Therefore, we find the word (w) so that this approximation holds good. We have to find a word to maximize the similarity of 'w', ii) $E_{Imatinib}-E_{CML}=1-(-1)^{*}=2$, $E_{Oxaliplatin}-E_{colorectal}=0.99-(-0.9)^{*}=1.89\sim2$. *Values taken from Table 1. Therefore, a GloVe draws a probability (P) and ratio (R), as shown in Table 2.

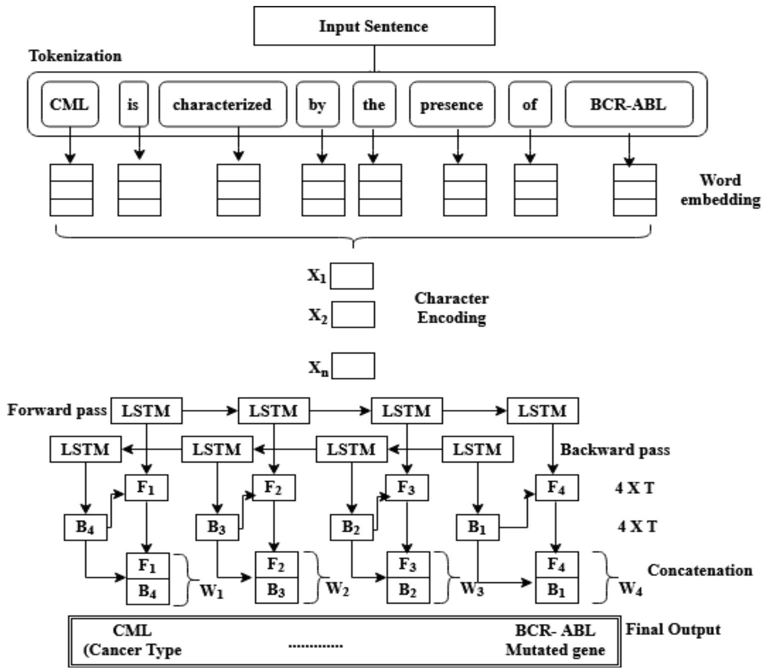


Figure 2. The architectural model for the proposed BiRNN network with LSTM cells for the NER task

Table 1. Dimensional vector representation for a given sample text

	Imatinib	CML	Nilotinib	Leukemia	Cancer	Oxaliplatin	Colorectal
Treat	1	-1	0.97	-0.95	-0.98	0.99	-0.9
Disease	0.09	0.93	0.01	0.99	0.98	0.02	0.97

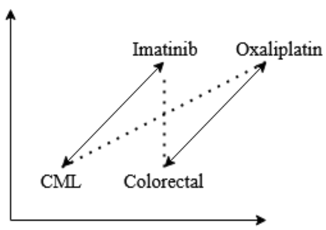


Figure 3. The 2D space representation of word vectors for GloVe word embedding

Here Table 2, P_iz/P_jz, indicating → p_iz=X_iz/X_i. P_iz: the probability of witnessing the words ‘i’ and ‘z’ jointly, calculated by dividing the total times ‘i’ and ‘z’ are seen together (X_iz) with the actual time ‘i’ is seen in the corpus (X_i). This allows drawing semantic similarities between the word-pairs in terms of probability and ratio of words, co-occurrences have higher values in the second row, extrapolating the disease names Let us consider two words, imatiniband oxaliplatin, and the word ‘z’ is: i) associated to Imatinib and not associated to oxaliplatin if P_iz/P_jz is greater than or equal to 1 and ii) associated to oxaliplatin and not associated to Imatinib if P_iz/P_ij is lesser than 1.

The GloVe approach will not recognize the target word if a word is not found in the dictionary. Further, for an out-of-vocabulary (OOV) word, the GloVe model simply assigns a random vector value. The character level embedding represents each word with a vector of numbers at the character level [26]. From a bird’s eye view, a sliding window moves the window character by character, making the next character prediction. As the window slides every time, the embedding concentrates on the sequence of multiple characters, capturing the information indicated by these characters. Lastly, all character-level embeddings are joined to form a word. The main drawback of language modelling is that character and word-level embedding assigns the same vector number for every appearance of the word/character in the context. This becomes complicated for heteronyms like ‘bank,’ and ‘orange’. So assigning the same vector will make the model inconsistent and create chaos in the network.

Table 2. Probability ratio between the target and the keyword

P & R	z=CML	z=colorectal
P(z|imitanib)	3.4	0.23
P(z|oxaliplatin)	0.66	1.83

2.3. Long-short-term-memory (LSTM) networks

LSTM, a variation of RNN model, learns a pattern even when the pattern is scattered across the full sentence [27]. The LSTM comprises four stages: i) eliminate stage, ii) input stage, iii) concatenation stage, and iv) output stage. Here, the vector representation of current and previous inputs is passed through activation functions tanh and sigmoid. The input layer performs a series of operations, making the values explode exponentially. With tanh activation, these values are squished between -1 to +1 [28]. The sigmoid activation function converts the values in the range 0 to 1 [29]. Each stage of LSTM is explained below in detail, Figure 4:

- Elimination stage: this takes the input from ‘k’ and ‘O_{t-1}’. For instance, the previous hidden state is ‘CML’ with the tag ‘cancer,’ and the present input is BCR-ABL with no label. The vector for BCR-ABL and the associated weight of previously seen ‘CML’ is passed into a sigmoid function. The output is then multiplied by the previous output O_{t-1}. If the value is closer to zero, then the state information is forgotten, and if it falls closer to 1, the state information is retained. Therefore, the tag ‘Cancer’ is ignored.
- Input stage: it takes the hidden state information with the weight‘ h_{t-1}’andthe vector representation of the present state ‘x_t’. In our example, the hidden weight of CML and vector representation of BCR-ABL is considered, indeed, multiplied by the tanh activation values of the previous hidden and present input state.

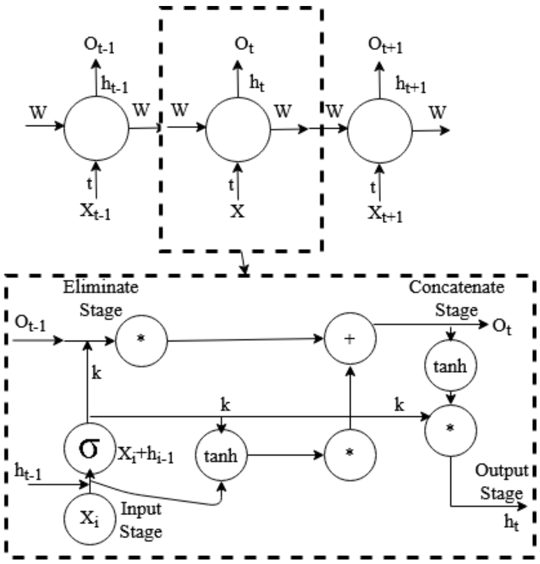


Figure. 4. The workflow of an LSTM network

The three stages of an RNN model are shown in the upper part. The lower part indicates the dissection of a single-stage, unfolding the four evaluation stages such as: i) input, ii) eliminate, iii) concatenation, and iv) output stage. Here, O_{t-1}: previous output, O_t: present output, σ: sigmoid activation function, tanh: tanh activation function, h_{t-1}: Previous hidden state, h_t: Present hidden state, X_t: Present input, k: Sigmoid value of X_t+h_{t-1}, *: point wise multiplication, +: point wise addition

- Concatenation stage: all the values are combined and form new information in an RNN memory unit. To do so, it concatenates the values of elimination and input stages and obtains a new value called O_t
- Output stage: the final stage reveals the tag to be marked for the current state information stored in the memory unit (concatenation stage). The O_t value is passed through the tanh activation function and multiplied with the sigmoid function of the present and previous hidden states. Therefore, BCR-ABL is

marked with the 'mutation' tag. In the backward pass, as the word 'gene' precedes the BCR-ABL, the labels are altered and finally fit the tag 'mutated gene' for BCR-ABL.

2.4. The flowchart and Pseudocode for Bi-directional RNN and LSTM based Bio-NER system

Figure 5 shows the flowchart of the entire process in a typical ML perspective. At first, the input data is fetched from the dataset, and then a series of ML actions take place to tag the associated tags for the features properly. The algorithmic procedure is indicated below in Algorithm 1. The algorithm illustrates how to extract the feature and tag appropriate labels.

Algorithm 1. Algorithm for feature extraction and bio-entity tagging using Bi-RNN-LSTM model

```
Input: 'in_sen' ➔ an input sentence for entity tagging
Output: 'out_sen ➔ the output with the bio-entities tagged (of the seven entity types)
Procedure:
 Remove stopwords and tokenize ['in_sen']
 For each word Wi in S, where, S={W1, W2, … Wn}
 For each character Ci in W, where, W={C1, C2, … Cn}
                  Draw probability ratio P and R (described in Table 2. under section: 2.2)
                  extract important features
                  Train features for LSTM model
 End For
            i) Calculate [h<t>], [x<t>] and [h<t-1>]
```

The input dataset is divided into training and test set. The essential features are extracted from the training data through GloVe embeddings at both character and word-level ('i' indicates the embedding model, as described in section 2.2). Further, the extracted features are passed as input for the Bi-RNN LSTM model ('ii' indicates algorithm 1). The model is trained, and features are learned. The test data is exposed for the learned model, and the prediction results are estimated using evaluation metrics, as shown in section 3.6.

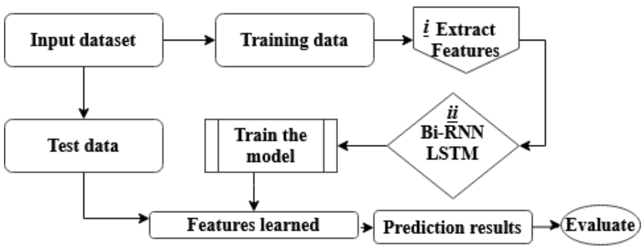


Figure 5. Flowchart of the neural network perspective for Bio-NER

3. RESULTS AND DISCUSSIONS

3.1. Datasets

Generally, a dataset is crucial in an implementation task containing 'enough' data for the training process and arrive at empirical hypotheses. Thus, in a data-driven learning mode, the accuracy is dependent on the type of corpus used. Initially, GENIA3.0 (source: www.nactem.ac.uk/genia/genia-corpus) was chosen for the training purpose; however, the tagging was not accurate as this corpus is suited for generic entities marked with the terms related to human blood lines and transcription factors. Therefore, 10,000 abstracts were selected randomly from MEDLINE using a PubMed search engine to have more specific medical terms. The abstracts were drawn with selective terms such as Cancer type, for instance, CML. There were a total of 1,550,970 words counted using a small python script. The sentences were then run through a python model called ScispaCy v0.2.4 (source: allenai.github.io/scispacy/) using a python code. The resulting tagged entities were tested against BioCreAtIvE II (source: https://biocreative.bioinformatics.udel.edu/tasks/biocreative-ii/) and GeneTag [30]. The entities were then compared, and wherever required, slight modifications were made manually to ensure the tagging is correctly completed.

3.2. Tools used for implementation

The proposed model is executed on the Spyder, v5.1.1 (https://www.spyder-ide.org/). Keras 2.4.0 is used for the neural network package (https://keras.io/), with TensorFlow v2.6.0(https://www.tensorflow.org/) running on top of the Keras application [31]. Python 3.9.0 (https://www.python.org/) is used for the coding. Keras is a neural network module that helps to design, fit, and assess a proposed model. An interesting fact about Keras is that it is more user-friendly as compared to Tensorflow. Keras library only provides low-level

APIs; however, it runs on Tensorflow (supports high-level APIs also); therefore, Keras supports both low and high-level APIs.

3.3. Data splitting method

The k-fold cross-validation (CV) approach is used to evaluate the performance of the model. First, the value of k is set to ten [32]. Next, the entire dataset is split into ten portions (for k=10). One part out of ten is reserved for the validation set. The remaining nine portions (k-1) are held for training, and the predictive performance is recorded at each evaluation stage. The process is repeated, changing the validation and test portions, and finally, the average of these measures is considered. The words are categorized into six entity classes: cancer type, genes, mutations, protein type, DNA structure name, and drugs and performed k-fold CV. The results are illustrated in Figure 6. Each parameter has ten recall, precision, and f-score values. The average of ten folds is calculated and presented in the Table 3.

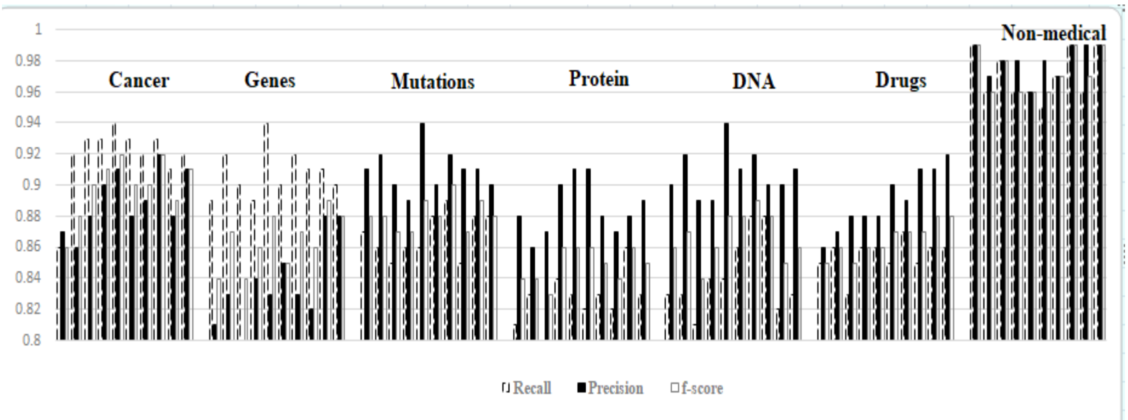


Figure 6. A graphical representation of 10-cross k-fold validation for seven parameters.

3.4. Baseline features

Keras is a popular deep learning framework that works efficiently with python code. The fascinating feature of Keras is that it consists of TensorFlow, CNTK, Theano, MXNet, and PlainML as the backend engines to support the deployment of deep learning mechanisms. The typical deep learning stages are defined in four layers in Keras; i) Dense layer: point-wise multiplication and the bias function is executed, ii) Activation layer: a transfer function to initiate the learning task, iii) Dropout layer: the layer is used to avoid the 'overfit' of the network through regularization techniques. During the training process, some percentage of layers are "dropped out" / left out. Thus, the model treats a layer differently based on the features of the previous layer [33], iv) Lambda layer: covers the inconsistent aspect of the learning and reduces the random fluctuations in the model.

3.5. Parameters setting

The total words in the input are the number of hidden layers in the network. The dropout value was initialized to 0.6 at the input layer and 0.5 at LSTM and softmax layers to address the overfitting problem of the network. Having fewer neurons often results in underfitting because the network will have insufficient data for training [34]. Therefore, 120 hidden neurons are used at each hidden layer. An optimizer must adjust the weights to avoid a high loss function value. Here, the AdaGrad optimizer learns the frequently occurring features, and the learning parameters are updated at every epoch. The learning rate and epsilon are set to 0.01 and 1e-08. The batch size=40, epoch=1000 with loss/error over training set indicated at each epoch execution.

3.6. Evaluation metrics

Figure 7 shows the final output obtained from the proposed model. It shows: i) the epochs and loss. In the beginning, the loss was equal to 67.3 and later reduced to 10.85 at the end of 1000 epochs; ii) input provided by the user; iii) entity-label tagged by the system. The LSTM model has removed all irrelevant words, and the tagging is performed only on keywords; iv) The cancer name and drug list. The performance of the output is evaluated through the following metrics; micro and macro F-measure alongside recall and precision scores [35]. This is because macro scores show how the system performs across all the data provided in the testing phase and micro score indicates the performance for every input. The other terminologies used are: i) true positive: the training and the test data are matched. For example, CML is a cancer type. It is identified as the Cancer type in the testing phase; ii) false positive: the entity identified in the testing phase is incorrect and does not match the entity type mentioned in the training phase. For example, GIST (tumor or Cancer type) is identified as mutation; iii) false negative: the entity type though present in the training data, is mentioned as not present. For example; SMARCA4 is not tagged as mutation though it is present in the training data; iv) precision indicates the correctly classified entities across all the entity types, for example, correctly classified number of Cancer type, genes, mutations, drug name and so on, over the actual number of individual entities in the training set; v) recall measures, for a particular entity type how many were correctly classified, for example: out of all cancer type instances, how many were correctly identified as a cancer type; vi) f-score calculates the average of both recall and precision; and vii) accuracy is the fraction of correctly classified entities over the total number of entities.

$$P = \frac{TP}{TP + FP} \tag{1}$$

$$R = \frac{TP}{TP + FN} \tag{2}$$

$$F = \frac{2 \; X \; R \; X \; P}{R + P} \tag{3}$$

$$MacroF = \frac{1}{N}\sum_{i=1}^{N} F_i \tag{4}$$

$$MacroP = \frac{1}{N}\sum_{i=1}^{N} P_i \tag{5}$$

$$MacroR = \frac{1}{N}\sum_{i=1}^{N} R_i \tag{6}$$

$$MicroF = \frac{2 \; X \; MicroP \; X \; MicroR}{MicroP + MicroR} \tag{7}$$

$$MicroP = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} TP_i + \sum_{i=1}^{N} FP_i} \tag{8}$$

$$MicroR = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} TP_i + \sum_{i=1}^{N} FN_i} \tag{9}$$

where N indicates the total entities types and P, R, F indicates precision, recall, and F-measure, and lastly, P_i, R_i, F_i indicates precision, recall, and F-score, for instance 'i'. The micro-scores for each of the entity types are presented in Table 3. The average is indicated in the table. The dataset was evaluated using a 10-fold CV as described in Figure 6. To calculate the macro scores, each entity-type score for a particular R, P, and F-score is enumerated by the total number of entity types (N=6). Therefore,

$$macroR = \frac{0.92 + 0.90 + 0.86 + 0.82 + 0.84 + 0.85}{6} = 0.86$$

$$macroP = \frac{0.89 + 0.83 + 0.91 + 0.88 + 0.90 + 0.89}{6} = 0.88$$

$$macroF = \frac{0.96 + 0.86 + 0.88 + 0.84 + 0.86 + 0.86}{6} = 0.87$$

3.7. Complexity

Each sentence (S_i) (considering the longest sentence) is compared against the training corpus of length (t_l) for an entity match. Thus, the local alignment complexity achieved is O ($S_i t_l$). Further, each sentence has multiple entity types, which are checked across all the test sentences of length (t_m). Therefore, for each entity pattern (e_p), the complexity will be O (e_p t_m t_l).

3.8. Comparison of existing research and current study

Table 4 demonstrates a comparison study between previous research works and the present study. Values in bold represent the overall best results. The highest score achieved in each row is indicated in bold. The last row values are the accuracy of the proposed model. Although the work [36] is better than the proposed model, only one entity phenotype for human-was considered. However, in the present research study, six different entities were considered for the tagging process. R: Recall; P: Precision; F: F-Score From a thorough insight of the authors', no research work adopted forward and backward pass along with LSTM variation. Due to this inclusion, this study witnessed elevated scores in the recall, precision, and f-score for the six entity types. Further, the character-based RNN is essential to identify the grammatically correct sequences for any morphological representation. With word-based RNN, the model displays high accuracy at a minimal computational time. Owing to these factors, the proposed model can mark complex and lengthy words such as epoxygenase and antitrypsin-glutathione. Besides, incorrectly marked entity tags would be corrected in the backward pass. There were no gazetteers involved for manually tagging, removing a tiresome workload of marking entity tags. These strategies helped achieve a remarkable improvement in the performance metrics.

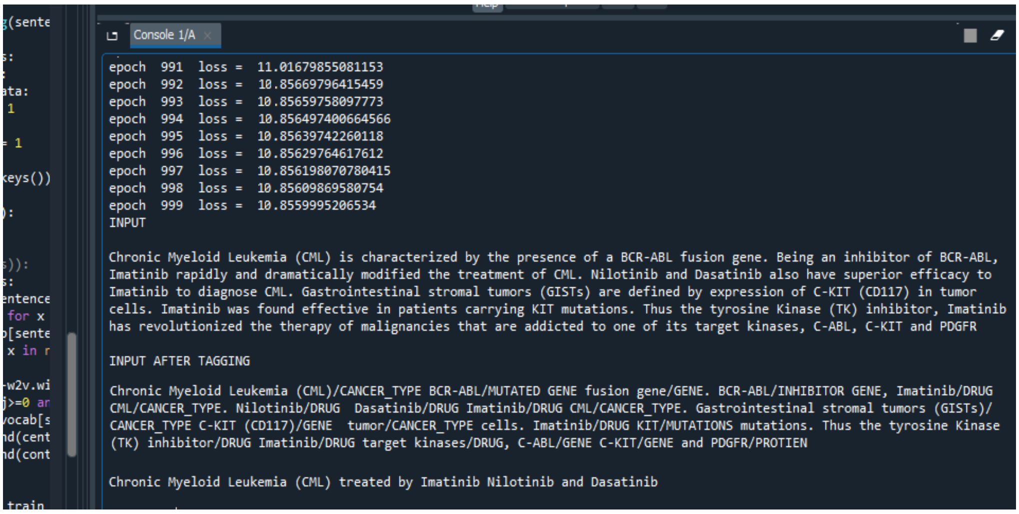


Figure. 7. The final output of the proposed system

Table 3. Micro-scores evaluated using [7]-[9] for each entity type

Entity Type–Entity Instances	Recall	Precision	F-Measure
Cancer Type-287926	0.92	0.89	0.90
Genes–156718	0.90	0.83	0.86
Mutations–379871	0.86	0.91	0.88
Protein Type-190171	0.82	0.88	0.84
DNA Structure Name–180275	0.84	0.90	0.86
Drugs 245689	0.85	0.89	0.86
Non-medical Words-110020	0.97	0.98	0.97

Table 4. A comparison study with the existing systems

State-of-art systems	R	P	F
Bio-NER using Deep Neural Networks [36]	76%	66%	71%
NE using Statistical NER [37]	79%	68%	73%
Bio-NER using RNN and LSTM [20]	73%	74%	86%
Bio-NER using rich features of CRFs for human phenotype terms [38]	87%	85%	86%
NER on Arabic text (only on the regular text) [39]	-	-	90.6%
Proposed model BiRNN using both forward and backward pass	86%	89%	88%

3.9. Discussion: challenges and future directions

DL-based Bio-NER significantly overcomes the disadvantages of the traditional NER models, such as feature engineering and pre-trained embeddings. However, like any other knowledge representation system, the RNN-based approach also suffers from challenges. The proposed model performs different pre-processing tasks such as tokenization, word and character embeddings. Due to this, the time complexity depends on these tasks plus an RNN training phase. To avoid this, the loss function could be presented after removing the stop words from the input, such as 'a', 'an', and so on. The observations could be ranked based on the frequency of words apart from stop words. One more possibility of reducing the time is considering the gradients of only the last iteration epoch rather than the entire sequence. The output at each step could be merged without worrying about the previous input sequence at different time steps. Further, the quality of annotated corpus plays a major role in performance. For example, BRCA2 is marked as a gene in GENETAG and mutation in BioCreAtiVE II corpus, creating confusion in entity labeling, and 17% of entities in GENIA are marked with duplicate entity tags. The same was observed even with the IL-2 gene. Out of 72 instances, 16 times, it was annotated as DNA and 56 times as protein. Some of the tags in GENETAG are not marked appropriately to their base category. For instance, T-cell and b-cells are not marked as cell-type, and 10 times t-cell occurred in the MEDLINE abstract and were marked as gene since the "T-cells and gene regulation" term appeared 48 times in the corpus. Thus, in the backward propagation, the t-cell was tagged as gene type and not as a cell type. T-cell receptor (TCR) gene was correctly marked as gene type; however, with such influence, even b-cell are marked as gene type since t-cell and b-cell appeared together in the corpus 108 times. These kinds of incorrect tagging increase the false-negative ratio. Protein molecules names (e.g., Dystrophin) were marked as DNA type. However, this error was reduced for some instances when the context information was increased (e.g., the length of input was increased by 1-fold). Further, names such as 'antigenic' were incorrectly identified as entity thereby increasing false positives. Although the pre-trained embeddings enabled the model to learn the context information precisely, there were 14089 errors (0.9%), which contributed to the overall misclassification.

Future enhancements: there are still many unanswered questions related to Bio-NER. For instance, it is still a question dealing with overlapping entity classes, such as cell lines vs. cell types and protein molecules vs. DNA molecules. Thus, in the future direction of the present study, the authors wish to perform boundary identification to handle overlapping entities and implement semantic parsing rules to identify the impact and relationship between different vital terms present in the input corpus. The human annotation could be reduced by applying ML techniques. Further, a catalog could be created to understand the samples of healthy vs. diseased individuals. Additionally, pipeline architecture could be designed to recognize entities and connect a feedback network to interact with different entity types, resolving the error propagation in further stages.

4. CONCLUSION

The bi-directional RNN LSTM technique is applied for biomedical and embedding at the word and character level in the present study. A total of 1,550,970 words from MEDLINE are used as a dataset. The GloVe model is put forward on these words for word-vector representation at both character and word levels. The LSTM component is created in the next step to interpret the tagging process with both forward and backward pass with the BiRNN model. Thus, with the LSTM, lengthy sequential data can be handled efficiently. From the experiments carried out, it is evident that the RNN model is better than the other traditional ML techniques such as SVM and HMM. The performance evaluation of the proposed system is judged through recall, precision, f-measure, and accuracy scores. The proposed approach outperforms the existing methodologies. Thus, the results of this study make the clinicians' task easy by providing an accurate and robust biomedical annotated corpus.

ACKNOWLEDGMENTS

Author Rashmi S expresses sincere gratitude to Science and Engineering Research Board (DST-SERB), New Delhi, India, for providing a research grant (NPDF, sanction order no PDF/2019/000254). In addition, the authors would like to thank the Department of Computational and Data Sciences, Indian Institute of Science, Bangalore, India, for extending the essential support required to carry out the study.

AVAILABILITY OF DATA AND MATERIALS

The necessary code and datasets are stored in the GitHub repository (https://github.com/RashmiSKarthik/Bio-NER). The researchers may replicate any methodology used in the present study with due citations. However, kindly write a mail to drrashmis64@gmail.com to obtain access to this repository.

REFERENCES

[1] T. Piliouras *et al.*, "Electronic health record systems: a current and future-oriented view," in *9th Annual Conference on Long Island Systems, Applications and Technology, LISAT 2013*, May 2013, pp. 1–6, doi: 10.1109/LISAT.2013.6578225.

[2] O. Fennelly *et al.*, "Successfully implementing a national electronic health record: a rapid umbrella review," *International Journal of Medical Informatics*, vol. 144, Dec. 2020, doi: 10.1016/j.ijmedinf.2020.104281.

[3] P. Sun, X. Yang, X. Zhao, and Z. Wang, "An overview of named entity recognition," in *Proceedings of the 2018 International Conference on Asian Language Processing, IALP 2018*, Nov. 2019, pp. 273–278, doi: 10.1109/IALP.2018.8629225.

[4] H. J. Song, B. C. Jo, C. Y. Park, J. D. Kim, and Y. S. Kim, "Comparison of named entity recognition methodologies in biomedical documents," *BioMedical Engineering Online*, vol. 17, Nov. 2018, doi: 10.1186/s12938-018-0573-6.

[5] J. Li, A. Sun, J. Han, and C. Li, "A survey on deep learning for named entity recognition," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2020, doi: 10.1109/tkde.2020.2981314.

[6] R. Phan, T. M. Luu, R. Davey, and G. Chetty, "Deep learning based biomedical NER framework," in *Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence, SSCI 2018*, Nov. 2019, pp. 33–40, doi: 10.1109/SSCI.2018.8628740.

[7] M. Dias, J. Boné, J. C. Ferreira, R. Ribeiro, and R. Maia, "Named entity recognition for sensitive data discovery in Portuguese," *Applied Sciences*, vol. 10, no. 7, Mar. 2020, doi: 10.3390/app10072303.

[8] D. Campos, S. Matos, and J. Luis, "Biomedical named entity recognition: a survey of machine-learning tools," in *Theory and Applications for Advanced Text Mining*, InTech, 2012.

[9] S. Rashmi, M. Hanumanthappa, and N. M. Jyothi, "Text-to-speech translation using support vector machine, an approach to find a potential path for human-computer speech synthesizer," in *Proceedings of the 2016 IEEE International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2016*, Mar. 2016, pp. 1311–1315, doi: 10.1109/WiSPNET.2016.7566349.

[10] S. Rashmi, M. Hanumanthappa, and M. V. Reddy, "Hidden markov model for speech recognition system—a pilot study and a Naive approach for speech-to-text model," in *Advances in Intelligent Systems and Computing*, vol. 664, Springer Singapore, 2018, pp. 77–90.

[11] I. Fernandes, H. L. Cardoso, and E. Oliveira, "Applying deep neural networks to named entity recognition in Portuguese texts," in *2018 5th International Conference on Social Networks Analysis, Management and Security, SNAMS 2018*, Oct. 2018, pp. 284–289, doi: 10.1109/SNAMS.2018.8554782.

[12] W. Saad, W. A. Shalaby, M. Shokair, F. A. El-Samie, M. Dessouky, and E. Abdellatef, "COVID-19 classification using deep feature concatenation technique," *Journal of Ambient Intelligence and Humanized Computing*, Mar. 2021, doi: 10.1007/s12652-021-02967-7.

[13] L. Yuwen, S. Chen, and X. Yuan, "G2Basy: a framework to improve the RNN language model and ease overfitting problem," *PLoS ONE*, vol. 16, Apr. 2021, doi: 10.1371/journal.pone.0249820.

[14] L. Li, L. Jin, Z. Jiang, D. Song, and D. Huang, "Biomedical named entity recognition based on extended recurrent neural networks," in *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Nov. 2015, pp. 649–652, doi: 10.1109/BIBM.2015.7359761.

[15] M. N. A. Ali and G. Tan, "Bidirectional encoder–decoder model for arabic named entity recognition," *Arabian Journal for Science and Engineering*, vol. 44, no. 11, pp. 9693–9701, Nov. 2019, doi: 10.1007/s13369-019-04068-2.

[16] H. Cho and H. Lee, "Biomedical named entity recognition using deep neural networks with contextual information," *BMC Bioinformatics*, vol. 20, no. 1, Dec. 2019, doi: 10.1186/s12859-019-3321-4.

[17] C. Lyu, B. Chen, Y. Ren, and D. Ji, "Long short-term memory RNN for biomedical named entity recognition," *BMC Bioinformatics*, vol. 18, no. 1, Dec. 2017, doi: 10.1186/s12859-017-1868-5.

[18] J. Li *et al.*, "WCP-RNN: a novel RNN-based approach for Bio-NER in Chinese EMRs," *The Journal of Supercomputing*, vol. 76, no. 3, pp. 1450–1467, Mar. 2020, doi: 10.1007/s11227-017-2229-x.

[19] S. Chowdhury *et al.*, "A multitask bi-directional RNN model for named entity recognition on Chinese electronic medical records," *BMC Bioinformatics*, vol. 19, Dec. 2018, doi: 10.1186/s12859-018-2467-9.

[20] Tian, Y., Shen, W., Song, Y., Xia, F., He, M., & Li, K. Improving biomedical named entity recognition with syntactic information. *BMC Bioinformatics*, 21(1), 2020, doi: 10.1186/s12859-020-03834-6.

[21] N. Perera, M. Dehmer, and F. Emmert-Streib, "Named entity recognition and relation detection for biomedical information extraction," *Frontiers in Cell and Developmental Biology*, vol. 8, Aug. 2020, doi: 10.3389/fcell.2020.00673.

[22] T. H. Yang, T. H. Tseng, and C. P. Chen, "Recurrent neural network-based language models with variation in net topology, language, and granularity," in *Proceedings of the 2016 International Conference on Asian Language Processing, IALP 2016*, Nov. 2017, pp. 71–74, doi: 10.1109/IALP.2016.7875937.

[23] N. Iqbal and N. Iqbal, "Imatinib: a breakthrough of targeted therapy in cancer," *Chemotherapy Research and Practice*, vol. 2014, pp. 1–9, May 2014, doi: 10.1155/2014/357027.

[24] T. Luo and H. Yang, "Two-layer neural networks for partial differential equations: optimization and generalization theory," Jun. 2020. *ArXiv:2006.15733*.

[25] P. Lauren, G. Qu, G.-B. Huang, P. Watta, and A. Lendasse, "A low-dimensional vector representation for words using an extreme learning machine," in *2017 International Joint Conference on Neural Networks (IJCNN)*, May 2017, pp. 1817–1822, doi: 10.1109/IJCNN.2017.7966071.

[26] M. Gridach, "Character-level neural network for biomedical named entity recognition," *Journal of Biomedical Informatics*, vol. 70, pp. 85–91, Jun. 2017, doi: 10.1016/j.jbi.2017.05.002.

[27] Y. Wang, "A new concept using LSTM neural networks for dynamic system identification," in *Proceedings of the American Control Conference*, May 2017, pp. 5324–5329, doi: 10.23919/ACC.2017.7963782.

[28] Y. Wang, Y. Li, Y. Song, and X. Rong, "The influence of the activation function in a convolution neural network model of facial expression recognition," *Applied Sciences*, vol. 10, no. 5, Mar. 2020, doi: 10.3390/app10051897.

[29] K. Hara and K. Nakayamma, "Comparison of activation functions in multilayer neural network for pattern classification," in *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, 1994, vol. 5, pp. 2997–3002, doi: 10.1109/ICNN.1994.374710.

[30] L. Tanabe, N. Xie, L. H. Thom, W. Matten, and W. J. Wilbur, "GENETAG: a tagged corpus for gene/protein named entity recognition," *BMC Bioinformatics*, vol. 6, May 2005, doi: 10.1186/1471-2105-6-S1-S3.
[31] R. Siddalingappa and S. Kanagaraj, "Anomaly detection on medical images using autoencoder and convolutional neural network," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 7, pp. 148–156, 2021, doi: 10.14569/IJACSA.2021.0120717.
[32] S. Yadav and S. Shukla, "Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification," in *Proceedings - 6th International Advanced Computing Conference, IACC 2016*, Feb. 2016, pp. 78–83, doi: 10.1109/IACC.2016.25.
[33] I. Bilbao and J. Bilbao, "Overfitting problem and the over-training in the era of data: particularly for artificial neural networks," in *2017 IEEE 8th International Conference on Intelligent Computing and Information Systems, ICICIS 2017*, Dec. 2017, vol. 2018-Janua, pp. 173–177, doi: 10.1109/INTELCIS.2017.8260032.
[34] I. Nusrat and S. B. Jang, "A comparison of regularization techniques in deep neural networks," *Symmetry*, vol. 10, no. 11, Nov. 2018, doi: 10.3390/sym10110648.
[35] A. Berger and S. Guda, "Threshold optimization for F measure of macro-averaged precision and recall," *Pattern Recognition*, vol. 102, Jun. 2020, doi: 10.1016/j.patcog.2020.107250.
[36] L. Yao, H. Liu, Y. Liu, X. Li, and M. W. Anwar, "Biomedical named entity recognition based on deep neutral network," *International Journal of Hybrid Information Technology*, vol. 8, no. 8, pp. 279–288, Aug. 2015, doi: 10.14257/ijhit.2015.8.8.29.
[37] P. Mishra, S. Biswas, and S. Dash, "Deep learning based biomedical named entity recognition systems," *Springer International Publishing*, 2020.
[38] M. Lobo, A. Lamurias, and F. M. Couto, "Identifying human phenotype terms by combining machine learning and validation rules," *BioMed Research International*, vol. 2017, pp. 1–8, 2017, doi: 10.1155/2017/8565739.
[39] I. El Bazi and N. Laachfoubi, "Arabic named entity recognition using deep learning approach," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 3, pp. 2025–2032, Jun. 2019, doi: 10.11591/ijece.v9i3.pp2025-2032.

BIOGRAPHIES OF AUTHORS

Rashmi Siddalingappa received B.Sc. and M.Sc. (Computer Science) and Ph.D. degree for phonetics and semantic analysis for natural language processing using data mining techniques from Bangalore University, Bengaluru, Karnataka, India. She is currently working as a national postdoctoral research fellow funded by SERB, India, in the computational and data sciences department at the Indian Institute of Science, Bengaluru, India. Her research interests include data mining, natural language processing, artificial intelligence, machine learning, and neural networks. She can be contacted at email: drrashmis64@gmail.com.

Kanagaraj Sekar received an M.Sc. and a Ph.D. in biophysics and crystallography from the University of Madras, India. His postdoctoral work pertained to protein crystallography, which he pursued at the Indian Institute of Science until 1992 and later at the Ohio State University from 1995. Since returning to India in 1998, he has been working at the Indian Institute of Science, Bengaluru, as a senior scientific officer of structural biology and bio-computing at its Bioinformatics Centre. During this period, he has held various positions, including those of a principal research scientist (2004–10) and an associate professor at the Computational and Data Sciences (CDS) department. Since 2016, he has been a professor at the CDS department and heads the Laboratory for Structural Biology and Bio-computing. He has been awarded the junior research fellowships of the Council of Scientific and Industrial Research (1984–88), the University Grants Commission of India (1988–89), and the senior research fellow of the CSIR (1989–92). In addition, the Department of Biotechnology of the Government of India awarded him the National Bioscience Award for Career Development, one of the highest Indian science awards, in 2004. His research in bioinformatics has covered protein crystallography, crystallographic and internet computing, and the development of value-added knowledge bases and algorithms. His studies have been documented by way of several articles, and Google Scholar has listed 211 of them. In addition, he has delivered keynote or plenary speeches at international seminars and conferences and has mentored many doctoral and postdoctoral scholars. He is also a member of the International Union of Crystallography (IUCr). He can be contacted at email: sekar@iisc.ac.in.

Image and video face retrieval with query image using convolutional neural network features

Imane Hachchane[1], Abdelmajid Badri[1], Aïcha Sahel[1], Ilham Elmourabit[1], Yassine Ruichek[2]
[1]Laboratoire d'Electronique, Energie, Automatique and Traitement de l'Information (EEA and TI), Faculté des Sciences et Techniques Mohammedia, Université Hassan II Casablanca, Mohammedia, Morocco
[2]IRTES-Laboratoire SET, Université de Technologie de Belfort Montbéliard, Belfor, France

Article Info

Article history:

Received May 7, 2021
Revised Dec 22, 2021
Accepted Dec 29, 2021

Keywords:

Classification
Convolutional neural network
Faster R-CNN
Image and video retrieval
Image processing
Image to video instance retrieval
Object recognition

ABSTRACT

This paper addresses the issue of image and video face retrieval. The aim of this work is to be able to retrieve images and/or videos of specific person from a dataset of images and videos if we have a query image of that person. The methods proposed so far either focus on images or videos and use hand crafted features. In this work we built an end-to-end pipeline for both image and video face retrieval where we use convolutional neural network (CNN) features from an off-line feature extractor. And we exploit the object proposals learned by a region proposal network (RPN) in the online filtering and re-ranking steps. Moreover, we study the impact of finetuning the networks, the impact of sum-pooling and max-pooling, and the impact of different similarity metrics. The results that we were able to achieve are very promising.

Corresponding Author:

Imane Hachchane
Laboratoire d'Electronique, Energie, Automatique and Traitement de l'Information (EEA and TI), Faculté des Sciences et Techniques Mohammedia, Université Hassan II Casablanca
Mohammedia, Morocco
Email: hachchaneimane@gmail.com

1. INTRODUCTION

The massive advances in internet technologies and the proliferation of smartphones, digital cameras and storage devices led to an increase in the popularity of visual search applications such as image retrieval, video retrieval or precisely instance search. By comparing a query against a database, instance search is used to extract images or videos of a particular object from large databases. It has been commonly used in product recognition, property identification, and other applications [1]–[3].

We should note that in one hand, image-to-image retrieval is a well-known field where large-scale face image retrieval has recently attracted attention, and a wide variety of methods have been proposed for face recognition and retrieval [4]–[7]. Following proper adaptation, well-known techniques for image retrieval were used for face recognition/retrieval, such as bag-of-visual words (BoVW). Other recent studies used convolutional neural network (CNN) for the feature extraction task [6].

On the other hand, image-to-video retrieval [8]–[10] is an asymmetric problem where the lack of temporal information in images stops us from using standard techniques for extracting video descriptors [11]–[14]. Traditionally, image-to-video retrieval techniques are based on a classic extraction methodes of hand-crafted features scale invariant feature transform (SIFT) [15], and binary robust independent elementary features (BRIEF) [16]. Smaller effort has been made to adapt deep learning techniques. We can apply standard features for image retrieval [17]–[20] by processing each frame as an independent image. More recent works showed that is possible to use CNN for feature extraction when working on videos [21], [22].

But not much work has been done in combining both, meaning having one pipeline for both image retrieval and video retrieval using one query image. Hence, in this paper, we investigate this issue. We are trying to retrieve the top N most relevant images and/or videos of an instance from a single image query instance. More specifically, we are working on face retrieval. In other words, giving an instance of a face in a query image, we are trying to retrieve the top N most relevant image instances and/or video instances from our database of videos and images of that specific face.

The main contribution of this paper is to build an end-to-end pipeline, for both image and video face retrieval using one query image. The pipeline takes advantage of off-the-shelf and fine-tuned features from an object detection CNN. We tested the impact of multiple similarity metrics, different network architectures, max-pooling and sum-pooling as well as the impact of most common reranking strategies.

2. RELATED WORK

Visual search and retrieval are in general an indexing and querying problem for visual data, which can be further divided into categories depending on the query type and database used. The most studied field in visual retrieval is image-to-image retrieval, where we use a query image to find the most relevant images from an image dataset [23], [24]. Generally speaking, visual search and retrieval remains an issue of indexing and querying visual data. This issue can be categorized depending on the type of queries and databases used. The most studied area in visual retrieval is image-to-image retrieval, were a we use a query image to retrieve the most relevant images from an image dataset [23], [24]. Another area of visual retrieval is video-to-video retrieval where a query video is used to retrieve relevant videos from a video dataset [25]. A further variant is video-to-image retrieval in which we use a query video to search a dataset of images [26], it is usually used in augmented reality. And of course we have the image-to-video retrieval where we search a database of videos using a query image [21]. In this paper, we merge two of those areas: Image-to-image retrieval and image-to-video retrieval. We focus on both image and video retrieval using one query image. More precisely, we are targeting face retrieval. Meaning, giving a query face image we are trying to retrieve the most relevant images and/or videos of that specific face.

Face retrieval is a difficult task because it is hard to adapt traditional image retrieval methodes (like bag of words) are difficult to apply to the field of face research [27]. Because the traditional descriptor based on the detection of key points (like SIFT) often fails due to the smooth surface of the face. Previous work, using a previously trained image classification convolutional neural network as a feature extractor, showed that it is more appropriate to use a fully connected layer for image retrieval [17]. Razavian *et al*. [28] Improved results by combining fully connected layers extracted from different image submatches. Later, the new work found that the convolutional layer is significantly better than the fully connected layer in image retrieval tasks [3], [28].

When working on image-to-image retrieval, a variety of CNN-based object detection pipelines have been proposed. In this paper, we are interested in Faster R-CNN [29], a CNN network created by Ren *et al*. They used a region proposal network (RPN) [30] in Faster R-CNN to remove the dependence of object propositions that exists in older CNN object detection systems. And, even though Faster R-CNN is designed to detect genral objects, Jiang and Learned-Miller [31] were able to highlight its impressive face detection performance, especially when retrained on a suitable face detection training set [6]. The current pipeline, that we are working on, uses off-the-shelf and finely tuned features of Faster R-CNN's end-to-end object detection architecture to extract global and local convolutional features in one pass and test their utility for image and video face retrieval using one query face image. We also test the impact of different similarity metrics, network architectures, max-pooling and sum-pooling, as well as reranking strategies.

3. METHODOLOGY

3.1. CNN-based representations

In our new pipeline, Figure 1, we examine the importance of using local and global CNN features extracted from pre-trained Faster R-CNN models [29] for image and video face retrieval. We use bounding boxs above our query images to define the instances that we are looking for. Faster R-CNN had two major parts that share a convolutional layer. The first one is RPN; it is a small neural network that glides over the last feature map of the convolution layers to predict whether an object is present or not, as well as the bounding box of those objects called windows. The second one is the classifier that learns to label each of those objects as one of the classes in the learning dataset [3].

As with earlier works [3], [32], and [33] our objective is to derive a compact image representation from Faster R-CNN activations. We construct the global descriptor by ignoring all of Faster R-CNN's layers that work with object propositions, and we derive features from the last convolutional layer. Taking the extracted activations of the convolution layer for an image or a frame into consideration, we group the activations of each filter to form an image descriptor with the same dimension as the number of filters in the convolution layer.

When working on constracting the local descriptor, the region pooling layer attached to the last convolutional layer is used to extract the convolutional activations for each of the object propositions gathered by the RPN for the local descriptor. This provides the capability of creating a local descriptor for every window proposal by aggregating the activations of that window in the RoI pooling layer. Sum-pooled features are l2-normalized in a manner similar to those described by several other authors [18], [32], followed by whitening and a second round of l2-normalization, while max-pooled features are only l2-normalized once without any whitening.

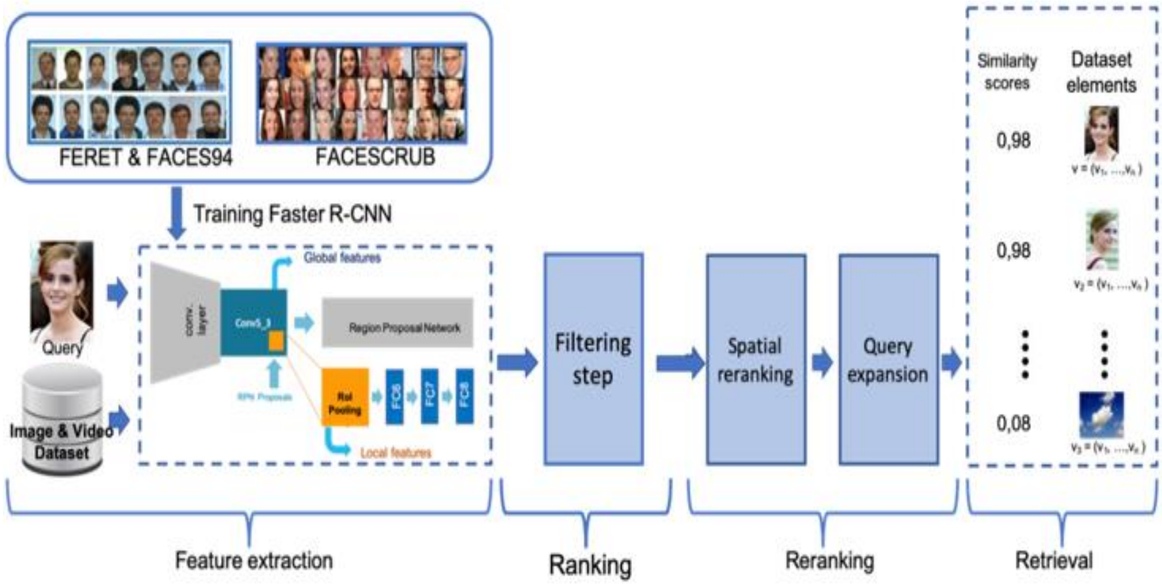


Figure 1. Proposed pipeline's architecture

3.2. Video and image retrieval

The feature extracting is done offline where we create the descriptors for the images, the video frames and the query images. At testing time (the online portion of the pipeline) we follow the raking strategies described in this section. We start with a filtering step, where the query features are compared to all the dataset items and then ranked using a similarity measure. At this step, we are still considering the entire frame as a query. After the filtering step, we locally analyze and re-rank the N upper elements. It is the spatial re-ranking. Last, we use query expansion (QE), in which we combine the descriptors of the M higher elements of the first ranking with the query descriptor to conduct a new search (M=5).

4. EXPERIMENTS

4.1. Utilized datasets

To test our methods, we need to use a dataset of images and videos. We could not find one, so we decided to merge two existing ones. These are the datasets we used:

- YouTube faces database [34]: The dataset contains 3,425 videos of 1,595 people, all of which were downloaded from YouTube. The database contains an average of 2.15 videos for each subject, with 48 frames being the shortest clip and 6,070 frames being the longest.
- FaceScrub [35]: 22,507 unconstrained face images amassed from the Internet. We added a framing box to the query images to surrounde the target faces.

The datasets we used to fine-tune the network:

- FERET [36]: This dataset has 3,528 images. We provide a framing box to the query images in order to surrounding the target faces.
- FACES94 [37]: This dataset has 2,809 images.

We also used the 55,127 unconstrained face images of the original FaceScrub dataset to fine-tune the network. When testing, we used 111 query images.

4.2. Experimental setup

According to previous works [3], [6], [21] deeper networks achieved better performance in extracting global and local features. Therefore, we decided to use the VGG16 architectures of Faster R-CNN and compare it with the ZF architecture to test the validity of the theory when working on both image and video retrieval. When working with the VGG16 architecture, the global descriptors are extracted from the last convolution layer "conv5_3" and are of dimension 512. And when working with the ZF architecture, the global descriptors are extracted from the last convolution layer "conv5" and are of dimension 256. For the local features, we group them from the Faster R-CNN RoI clustering layer. The global descriptors for the VGG16 architecture are extracted from the last convolution layer "conv5_3" and are of dimension 512, while the global descriptors for the ZF architecture are extracted from the last convolution layer "conv5" and are of dimension 256. We group local features using the Faster R-CNN region of interest (RoI) clustering layer.

We also experimented with widely used similarity metrics to see which one is more suitable for our pipeline. We tested the following similarity metrics: Cosine similarity metric, Euclidien similarity metric, Manhatan similarity metric, Chebychev similarity metric, Minkowski similarity metric, Canberra similarity metric, and Corrolation similarity metric. The following specifications were used for the experiments: Processor: Intel(R) Core (TM) i7-7700K CPU 4.20 GHz, RAM: 16 GB, OS: Ubuntu 16.04, Graphics card: NVIDIA GeForce GTX 1070.

We should note the extraction time for the VGG16 required an average of 16h 11min 30 seconds compared to an average of 7h 34min and 22 seconds when using ZF. This time difference can be explained by the sizes of the networks. The ranking took on average 2 seconds per query image; the re-ranking took an average of 16 seconds per query image, and when using the QE, the re-ranking took an average of 17 seconds per query image.

4.3. Off-the-shelf CNN features

In this section we evaluate using Faster R-CNN features for face image and/or video retrieval. We have tested different similarity metrics, as detailed above. The results, displayed in Table 1, were similar and close, but the best results were obtained using the cosine and the euclidien similarity metrics combined with our re-ranking strategies with a precision of 55.4%. But with the other similarity metrics, the query expansion and the spatial reranking did not improve the results.

Moreover, a comparative study of the sum and max pooling strategies of image-wise and region-wise descriptors was also conducted, with the results summarized in Table 1. Sum-pooling is better than max-pooling, according to our tests. It also confirms that Faster R-CNN with a VGG16 architecture trained on pascal VOC datasets performed best, which is consistent with previous research that had demonstrated that deep networks could deliver better results when extracting global and local features.

4.4. Fine-tuning the CNN

More importantly, we investigated the effects of fine-tuning a pre-trained network on recovery performance with the query objects to retrieve. We used the model VGG16 of Faster R-CNN pre-trained with the pascal VOC objects. We refined it using two datasets:

– We refined the first network with FERET and Faces94 datasets and we called it VGG16 (Feret and Faces94). Because of their small size, the Feret and the Faces94 datasets were combined, and the network's output layer was modified to return 422 class probabilities and their corresponding bounding box coordinates [6] (the 422 counts for the 269 classes in the FERET dataset and the 152 classes in the Faces94 dataset, plus one additional class for the background).
– We refined the second network with using the FaceScrub dataset. We called it VGG16 (Facescrub). For this network the output layer was modified to return 530 class probabilities and their corresponding bounding box coordinates (530 classes, plus one additional class for the background).

The initial parameters of Faster R-CNN as described in [19] did not change, but due to a reduced number of training samples, the number of iterations was reduced from 80,000 to 20,000. We use the refined networks of the tuning strategy (VGG16 (Feret and Faces94) and VGG16 (Facescrub)) on our image and video dataset to extract the descriptors and perform image and video face retrieval. Those results are presented in Table 2. This time the Manhattan similarity metric, also called city block, produced the best results. We should also note that the query expansion and spatial reranking slightly improved the results. When comparing the sum-pooling strategie to the max-pooling strategie of the image-wise and region-wise descriptors, sum-pooling gave better results than max-pooling with most similarity metrics. But max-pooling gave the best results when used with the Manhattan similarity metric with an accuracy of 76.2%.

We also compared different Faster R-CNN architectures trained on different datasets. We determined that deeper networks gave better results, which is consistent with the literature. We also noticed the datasets, on which the network was previously trained, had the most impact on the results. As we can see, when working with off-the-shelf networks, the networks trained on pascal VOC gave average results. But the best results were obtained when working with the networks trained for face classification, meaning trained on Fasecrub and Feret and Faces94 in our case. On that account, the VGG16 trained on Facescrub gave the best results because the nature of the photos in this dataset is more similar to the dataset that we are working on. Feret and Faces94 images were taken in a controlled environment, but Fasecrub images were amassed from the web and showcase the subject in different positions with different lighting setups and facial expressions which is closest to what videos can be. That is why the VGG16 trained on Facescrub gave the best results when used for retrieving face images and videos from a dataset of images and videos using one query image with a precision of 76.2%. So, we were able to improve the results with 13.7%.

Table 1. Mean average precision (mAP) of pre-trained Faster R-CNN models trained with microsoft COCO or pascal VOC

Metrics	Models	Pooling	Ranking	Re-ranking	QE
Cosine similarity metric	VGG16 (Pascal VOC)	sum	<u>0.551</u>	<u>0.551</u>	<u>0.554</u>
		max	0.538	0.545	0.544
	VGG16 (Microsoft COCO)	sum	0.545	0.521	0.516
		max	0.524	0.525	0.522
	ZF (Pascal VOC)	sum	0.550	0.539	0.538
		max	0.534	0.544	0.540
Euclidien similarity metric	VGG16 (Pascal VOC)	sum	<u>0.551</u>	<u>0.551</u>	<u>0.554</u>
		max	0.538	0.545	0.544
	VGG16 (Microsoft COCO)	sum	0.545	0.521	0.516
		max	0.524	0.525	0.522
	ZF (Pascal VOC)	sum	0.550	0.539	0.538
		max	0.534	0.544	0.540
Manhatan similarity metric	VGG16 (Pascal VOC)	sum	0.550	0.550	0.545
		max	0.540	0.543	0.538
	VGG16 (Microsoft COCO)	sum	0.543	0.513	0.507
		max	0.527	0.529	0.526
	ZF (Pascal VOC)	sum	0.547	0.535	0.530
		max	0.538	0.549	0.546
Chebychev similarity metric	VGG16 (Pascal VOC)	sum	0.497	0.482	0.493
		max	0.470	<u>0.451</u>	0.469
	VGG16 (Microsoft COCO)	sum	0.513	0.465	0.487
		max	0.488	0.437	0.453
	ZF (Pascal VOC)	sum	0.518	0.515	0.517
		max	0.499	0.459	0.490
Minkowski similarity metric	VGG16 (Pascal VOC)	sum	<u>0.551</u>	<u>0.551</u>	0.544
		max	0.538	0.545	0.544
	VGG16 (Microsoft COCO)	sum	0.545	0.521	0.516
		max	0.524	0.525	0.522
	ZF (Pascal VOC)	sum	0.550	0.544	0.536
		max	0.534	0.544	0.540
Canberra similarity metric	VGG16 (Pascal VOC)	sum	0.547	0.544	0.539
		max	0.528	0.516	0.518
	VGG16 (Microsoft COCO)	sum	0.538	0.516	0.512
		max	0.526	0.524	0.524
	ZF (Pascal VOC)	sum	0.540	0.538	0.537
		max	0.524	0.530	0.521
Corrolation similarity metric	VGG16 (Pascal VOC)	sum	<u>0.551</u>	<u>0.551</u>	0.544
		max	0.539	0.549	0.548
	VGG16 (Microsoft COCO)	sum	0.545	0.520	0.524
		max	0.524	0.522	0.517
	ZF (Pascal VOC)	sum	0.549	0.544	0.545
		max	0.537	0.542	0.537

4.5. Comparison

In this section we present a comparative study between our results and other results obtained using fisher vector (FV) and bag of visual word (BOVW). When working on video retrieval and image and video retrieval, our pipeline, which utilizes raw faster R-CNN features, outperformed all other techniques. The results are displayed in Table 3.

Table 2. Mean average precision (mAP) of the fine-tuned Faster R-CNN models with VGG16 architectures fine-tuned with Facescrub or Feret and Faces9 respectively

Metrics	Models	Pooling	Ranking	Re-ranking	QE
Cosine similarity metric	VGG16(Facescrub).	sum	0.757	0.737	0.706
		max	0.738	0.731	0.756
	VGG16(Feret and Faces94)	sum	0.577	0.570	0.563
		max	0.554	0.564	0.572
Euclidien similarity metric	VGG16(Facescrub).	sum	0.757	0.737	0.706
		max	0.738	0.731	0.756
	VGG16(Feret and Faces94)	sum	0.577	0.570	0.563
		max	0.554	0.564	0.572
Manhatan similarity metric	VGG16(Facescrub).	sum	0.738	0.695	0.734
		max	0.750	0.746	0.762
	VGG16(Feret and Faces94)	sum	0.565	0.561	0.553
		max	0.562	0.573	0.580
Chebychev similarity metric	VGG16(Facescrub).	sum	0.545	0.555	0.562
		max	0.564	0.579	0.605
	VGG16(Feret and Faces94)	sum	0.504	0.513	0.514
		max	0.495	0.501	0.500
Minkowski similarity metric	VGG16(Facescrub).	sum	0.757	0.727	0.747
		max	0.738	0.731	0.756
	VGG16(Feret and Faces94)	sum	0.577	0.570	0.560
		max	0.554	0.564	0.572
Canberra similarity metric	VGG16(Facescrub).	sum	0.742	0.742	0.760
		max	0.723	0.731	0.737
	VGG16(Feret and Faces94)	sum	0.567	0.569	0.568
		max	0.556	0.558	0.552
Corrolation similarity metric	VGG16(Facescrub).	sum	0.757	0.728	0.749
		max	0.741	0.731	0.748
	VGG16(Feret and Faces94)	sum	0.577	0.570	0.563
		max	0.557	0.568	0.573

Table 3. Comparative study with other techniques. Results provided as mAP

Method	YouTube Faces Database+Facescrub (an image and video dataset)	YouTube Faces Database (a video dataset)	FERET (an image dataset)
Our pipeline	0.762	0.903	0.8913
Faster R-CNN features+FV [21]	0.006	0.006	-
Faster R-CNN features+BOVW [21]	-	0.001	-
Log ICA II+KNN [38]	-	-	0.3553
Log ICA I+KNN [38]	-	-	0.3608
LGHP descriptor [7]	-	-	0.5460

5. CONCLUSION

In this paper, we demonstrate how to use CNN features from an object detection network for image and video face retrieval using one query image. We used Faster R-CNN features as our global and local descriptors in our end-to-end pipeline. We demonstrated that the best similarity metric to use with the off-the-shelf feature is the cosine similarity metric, and that the best one to use with refined networks is the Manhattan similarity metric. We also found that sum-pooling generally performs better, but when using the fine-tuned networks with the Manhattan similarity metrics, max-pooling gave the best results. We established that reranking strategies can improve the results. Most importantly, we proved that finetuned networks give the best results. So, when working on image and video face retrieval using one query image, we found the best results were obtained using a fine-tuned network combined with max-pooling, all our reranking strategies and using the Manhattan similarity metric. We determined that Finetuned CNN feature can give great results (76,2%) in real time (17 seconds per query image) when working on image and video face retrieval using a query image.

ACKNOWLEDGEMENTS

This work falls within the scope of Big Data and Connected Object (BDCO). We would like to thank the Hassan II University of Casablanca for financing this project.

REFERENCES

[1] D. Feng, M.-G. Liang, F. Gao, Y.-C. Huang, X.-F. Zhang, and L.-Y. Duan, “Towards large-scale object instance search: A multi-

block N-ary Trie," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 1, pp. 372–386, Jan. 2021, doi: 10.1109/TCSVT.2020.2966541.
[2] S. S. Tsai *et al.*, "Mobile product recognition," in *Proceedings of the international conference on Multimedia-MM '10*, 2010, Art. no. 1587, doi: 10.1145/1873951.1874293.
[3] A. Salvador, X. Giro-i-Nieto, F. Marques, and S. Satoh, "Faster R-CNN features for instance search," in *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2016, pp. 394–401, doi: 10.1109/CVPRW.2016.56.
[4] C.-W. Lin and S. Hong, "High-order histogram-based local clustering patterns in polar coordinate for facial recognition and retrieval," *The Visual Computer*, Mar. 2021, doi: 10.1007/s00371-021-02102-9.
[5] F.-C. Lin, H.-H. Ngo, and C.-R. Dow, "A cloud-based face video retrieval system with deep learning," *The Journal of Supercomputing*, vol. 76, no. 11, pp. 8473–8493, Nov. 2020, doi: 10.1007/s11227-019-03123-x.
[6] I. Hachchane, A. Badri, A. Sahel, and Y. Ruichek, "New faster R-CNN neuronal approach for face retrieval," in *Lecture Notes in Networks and Systems*, Springer International Publishing, 2019, pp. 113–120.
[7] S. R. Dubey, "Local directional relation pattern for unconstrained and robust face retrieval," *Multimedia Tools and Applications*, vol. 78, no. 19, pp. 28063–28088, Oct. 2019, doi: 10.1007/s11042-019-07908-3.
[8] L. Liu, J. Li, L. Niu, R. Xu, and L. Zhang, "Activity image-to-video retrieval by disentangling appearance and motion," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, pp. 2145–2153.
[9] J. Pont-Tuset, P. Arbelaez, J. T. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping for image segmentation and object proposal generation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 1, pp. 128–140, Jan. 2017, doi: 10.1109/TPAMI.2016.2537320.
[10] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 3431–3440, doi: 10.1109/CVPR.2015.7298965.
[11] A. F. De Araujo, "Large-scale video retrieval using image queries a dissertation submitted to the department of electrical engineering and the committee on graduate studies of stanford university in partial fulfillment of the requirements for the degree of doctor of philos," 2016.
[12] Joe Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 4694–4702, doi: 10.1109/CVPR.2015.7299101.
[13] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," Jun. 2014, [Online]. Available: http://arxiv.org/abs/1406.2199.
[14] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, pp. 4489–4497, doi: 10.1109/ICCV.2015.510.
[15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004, doi: 10.1023/B:VISI.0000029664.99615.94.
[16] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2010, pp. 778–792.
[17] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," Springer International Publishing, 2014, pp. 584–599.
[18] Y. Kalantidis, C. Mellina, and S. Osindero, "Cross-dimensional weighting for aggregated deep convolutional features," in *Lecture Notes in Computer Science*, Springer International Publishing, 2016, pp. 685–701.
[19] A. S. Razavian, J. Sullivan, S. Carlsson, and A. Maki, "Visual instance retrieval with deep convolutional networks," *ITE Transactions on Media Technology and Applications*, vol. 4, no. 3, pp. 251–258, 2016, doi: 10.3169/mta.4.251.
[20] L. Wu, Y. Wang, Z. Ge, Q. Hu, and X. Li, "Structured deep hashing with convolutional neural networks for fast person re-identification," *Computer Vision and Image Understanding*, vol. 167, pp. 63–73, Feb. 2018, doi: 10.1016/j.cviu.2017.11.009.
[21] I. Hachchane, A. Badri, A. Sahel, and Y. Ruichek, "Large-scale image-to-video face retrieval with convolutional neural network features," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 9, no. 1, pp. 40–45, Mar. 2020, doi: 10.11591/ijai.v9.i1.pp40-45.
[22] C. Zhang, B. Hu, Y. Suo, Z. Zou, and Y. Ji, "Large-scale video retrieval via deep local convolutional features," *Advances in Multimedia*, vol. 2020, pp. 1–8, Jun. 2020, doi: 10.1155/2020/7862894.
[23] L. Zheng, Y. Yang, and Q. Tian, "SIFT meets CNN: A decade survey of instance retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 5, pp. 1224–1244, May 2018, doi: 10.1109/TPAMI.2017.2709749.
[24] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas, "Query specific rank fusion for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 4, pp. 803–815, Apr. 2015, doi: 10.1109/TPAMI.2014.2346201.
[25] S. Poullot, S. Tsukatani, A. Phuong Nguyen, H. Jégou, and S. Satoh, "Temporal matching kernel with explicit feature maps," in *Proceedings of the 23rd ACM international conference on Multimedia*, Oct. 2015, pp. 381–390, doi: 10.1145/2733373.2806228.
[26] D. M. Chen and B. Girod, "A hybrid mobile visual search system with compact global signatures," *IEEE Transactions on Multimedia*, vol. 17, no. 7, pp. 1019–1030, Jul. 2015, doi: 10.1109/TMM.2015.2427744.
[27] C. Herrmann and J. Beyerer, "Fast face recognition by using an inverted index," in *Procedings SPIE 9405, Image Processing: Machine Vision Applicationd VIII, 940507*, Feb. 2015, Art. no. 940507, doi: 10.1117/12.2078988.
[28] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Jun. 2014, pp. 512–519, doi: 10.1109/CVPRW.2014.131.
[29] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.
[30] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.
[31] H. Jiang and E. Learned-Miller, "Face detection with the faster R-CNN," in *2017 12th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2017)*, May 2017, pp. 650–657, doi: 10.1109/FG.2017.82.
[32] A. B. Yandex and V. Lempitsky, "Aggregating local deep features for image retrieval," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, pp. 1269–1277, doi: 10.1109/ICCV.2015.150.
[33] G. Tolias, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of CNN activations," *Computer Vision and Pattern Recognition*, Nov. 2015, [Online]. Available: http://arxiv.org/abs/1511.05879.
[34] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *CVPR 2011*, Jun. 2011, pp. 529–534, doi: 10.1109/CVPR.2011.5995566.

[35] H.-W. Ng and S. Winkler, "A data-driven approach to cleaning large face datasets," in *2014 IEEE International Conference on Image Processing (ICIP)*, Oct. 2014, pp. 343–347, doi: 10.1109/ICIP.2014.7025068.
[36] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, "The FERET database and evaluation procedure for face-recognition algorithms," *Image and Vision Computing*, vol. 16, no. 5, pp. 295–306, Apr. 1998, doi: 10.1016/S0262-8856(97)00070-X.
[37] D. L. Spacek, "Faces94 a face recognition dataset," 2007.
[38] M. K. Bhowmik, P. Saha, A. Singha, D. Bhattacharjee, and P. Dutta, "Enhancement of robustness of face recognition system through reduced gaussianity in Log-ICA," *Expert Systems with Applications*, vol. 116, pp. 96–107, Feb. 2019, doi: 10.1016/j.eswa.2018.08.047.

BIOGRAPHIES OF AUTHORS

Imane Hachchane is a Ph.D. student in Image processing at the EEA and TI Laboratory, Hassan II University Casablanca, Faculty of Sciences and Technology of Mohammedia (FSTM) in Morocco. She received her Software Engineering Degree from the National School of Applied Sciences of Kenitra, Morocco in 2016. She's currently working on Facial Large Scale Image Retrieval under the supervision of Pr. A. Badri. Her main research interest is to enhance the accuracy and speed of largescale image and video face retrieval using neural networks and deep learning. She can be contacted at email: hachchaneimane@gmail.com.

Abdelmajid Badri is a holder of a doctorate in Electronics and Image Processing in 1992 at the University of Poitiers–France. In 1996, he obtained the diploma of the authorization to Manage Researches (Habilitation à Diriger des Recherches: HDR) to the University of Poitiers–France, on the image processing. He is a director at the Higher School of Technology (EST) at Casablanca and he is a University Professor (PES-C) at the University Hassan II-Casablanca-Morocco (FSTM). He is a member of the laboratory EEA and TI (Electronics, Energy, Automatic and information Processing) which he managed since 1996. He managed several doctoral theses. He is a co-author of several national and international publications. He is responsible for several research projects financed by the ministry or by the industrialists. He was member of several committees of programs of international conferences and president of three international congresses in the same domain. He is a member and co-responsible in several scientific associations in touch with his domain of research. He can be contacted at email: abdelmajid_badri@yahoo.fr.

Aïcha Sahel is a holder of a doctorate in Electronics and Image Processing in 1996 at the University of Poitiers-France. She is a university Professor at the University Hassan II-Casablanca-Morocco (FSTM) She is a member of the laboratory EEA and TI. The research works of A. Sahel concern the Communication and Information Technology (Electronics Systems, Signal/Image Processing and Telecommunication). She co-supervises doctoral theses and she is a co-author of several national and international publications. She is a member in financed research projects. She was a member of steering committees of three international congresses in the same domain of research. She can be contacted at email: sahel_ai@yahoo.fr.

Ilham Elmourabit is a holder of a doctorate in Telecommunication and information engineering in 2011 at the University Hassan II-Casablanca-Morocco (FSTM). She is a university Professor at the Hassan II University Casablanca, Faculty of Sciences and Technology of Mohammedia (FSTM) in Morocco. She is a member of the laboratory EEA and TI. The research works of I. Elmourabit concern the Communication and Information Technology. She co-supervises doctoral theses and she is a co-author of several national and international publications. She can be contacted at email: elmourabit.ilham@gmail.com.

Yassine Ruichek (Senior Member, IEEE) received the Ph.D. degree in control and computer engineering and the Habilitation à Diriger des Recherches (HDR) degree in physic science from the University of Lille, France, in 1997 and 2005, respectively. Since 2007, he has been a Full Professor with the University of Technology of Belfort-Montbéliard (UTBM). His research interests include computer vision, image processing and analysis, pattern recognition, data fusion, and localization, with applications in intelligent transportation systems and video surveillance. He can be contacted at email: yassine.ruichek@utbm.fr.

Paper's title should be the fewest possible words that accurately describe the content of the paper (Center, Bold, 16pt)

Abdel-Rahman Hedar[1,2], Patricia Melin[3], Kennedy Okokpujie[4] (10 pt)
[1]Department of Computer Science, Faculty of Computers & Information, Assiut University, Assiut, Egypt (8 pt)
[2]Department of Computer Science in Jamoum, Umm Al-Qura University, Makkah, Saudi Arabia
[3]Division of Graduate Studies, Tijuana Institute of Technology, Tijuana, Mexico
[4]Department of Electrical and Information Engineering, College of Engineering, Covenant University, Ogun State, Nigeria

Article Info

Article history:

Received month dd, yyyy
Revised month dd, yyyy
Accepted month dd, yyyy

Keywords:

First keyword
Second keyword
Third keyword
Fourth keyword
Fifth keyword

ABSTRACT (10 PT)

An abstract is often presented separate from the article, so it must be able to stand alone. A well-prepared abstract enables the reader to identify the basic content of a document quickly and accurately, to determine its relevance to their interests, and thus to decide whether to read the document in its entirety. The abstract should be informative and completely self-explanatory, provide a clear statement of the problem, the proposed approach or solution, and point out major findings and conclusions. **The Abstract should be 100 to 200 words in length.** References should be avoided, but if essential, then cite the author(s) and year(s). Standard nomenclature should be used, and non-standard or uncommon abbreviations should be avoided, but if essential they must be defined at their first mention in the abstract itself. No literature should be cited. The keyword list provides the opportunity to add 5 to 7 keywords, used by the indexing and abstracting services, in addition to those already present in the title (9 pt).

Corresponding Author:

Kennedy Okokpujie
Department of Electrical and Information Engineering, College of Engineering, Covenant University
Km. 10 Idiroko Road, Canaan Land, Ota, Ogun State, Nigeria
Email: kennedy.okokpujie@covenantuniversity.edu.nga

1. INTRODUCTION (10 PT)

The main text format consists of a flat left-right columns on A4 paper (quarto). The margin text from the left and top are 2.5 cm, right and bottom are 2 cm. The manuscript is written in Microsoft Word, single space, Time New Roman 10 pt, and maximum 12 pages for original research article, or maximum 16 pages for review/survey paper, which can be downloaded at the website: http://ijai.iaescore.com.

A title of article should be the fewest possible words that accurately describe the content of the paper. The title should be succinct and informative and no more than about 12 words in length. Do not use acronyms or abbreviations in your title and do not mention the method you used, unless your paper reports on the development of a new method. Titles are often used in information-retrieval systems. Avoid writing long formulas with subscripts in the title. Omit all waste words such as "*A study of ...*", "*Investigations of ...*", "*Implementation of ...*", "*Observations on ...*", "*Effect of.....*", "*Analysis of ...*", "Design of…", etc.

A concise and factual abstract is required. The abstract should state briefly the purpose of the research, the principal results and major conclusions. An abstract is often presented separately from the article, so it must be able to stand alone. For this reason, References should be avoided, but if essential, then cite the author(s) and year(s). Also, non-standard or uncommon abbreviations should be avoided, but if essential they must be defined at their first mention in the abstract itself. Immediately after the abstract, provide a maximum of 7 keywords, using American spelling and avoiding general and plural terms and multiple concepts (avoid, for example, 'and', 'of'). Be sparing with abbreviations: only abbreviations firmly established in the field may be eligible. These keywords will be used for indexing purposes.

Indexing and abstracting services depend on the accuracy of the title, extracting from it keywords useful in cross-referencing and computer searching. An improperly titled paper may never reach the audience for which it was intended, so be specific.

The Introduction section should provide: i) a clear background, ii) a clear statement of the problem, iii) the relevant literature on the subject, iv) the proposed approach or solution, and v) the new value of research which it is innovation (within 3-6 paragraphs). It should be understandable to colleagues from a broad range of scientific disciplines. Organization and citation of the bibliography are made in Institute of Electrical and Electronics Engineers (IEEE) style in sign [1], [2] and so on. The terms in foreign languages are written italic (*italic*). The text should be divided into sections, each with a separate heading and numbered consecutively [3]. The section or subsection headings should be typed on a separate line, e.g., 1. INTRODUCTION. A full article usually follows a standard structure: **1. Introduction, 2. The Comprehensive Theoretical Basis and/or the Proposed Method/Algorithm** *(optional)*, **3. Method, 4. Results and Discussion, and 5. Conclusion.** The structure is well-known as **IMRaD** style.

Literature review that has been done author used in the section "INTRODUCTION" to explain the difference of the manuscript with other papers, that it is innovative, it are used in the section "METHOD" to describe the step of research and used in the section "RESULTS AND DISCUSSION" to support the analysis of the results [2]. If the manuscript was written really have high originality, which proposed a new method or algorithm, the additional section after the "INTRODUCTION" section and before the "METHOD" section can be added to explain briefly the theory and/or the proposed method/algorithm [4].

2. METHOD (10 PT)

Explaining research chronological, including research design, research procedure (in the form of algorithms, Pseudocode or other), how to test and data acquisition [5]–[7]. The description of the course of research should be supported references, so the explanation can be accepted scientifically [2], [4]. Figures 1-2 and Table 1 are presented center, as shown below and cited in the manuscript [5], [8]–[13]. The settlement curves produced at SG1 has been illustrated in Figure 2(a) and SG2 has been illustrated Figure 2(b).

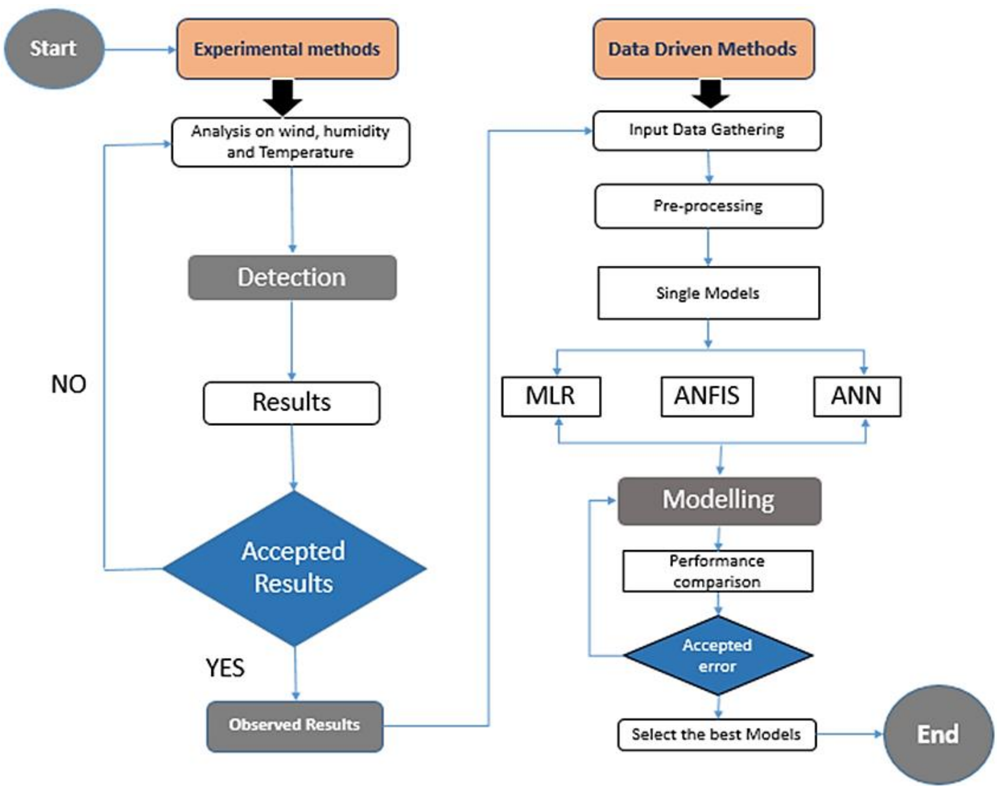
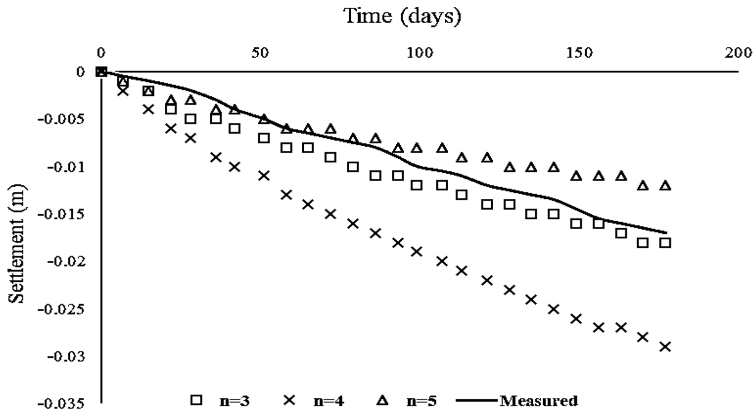
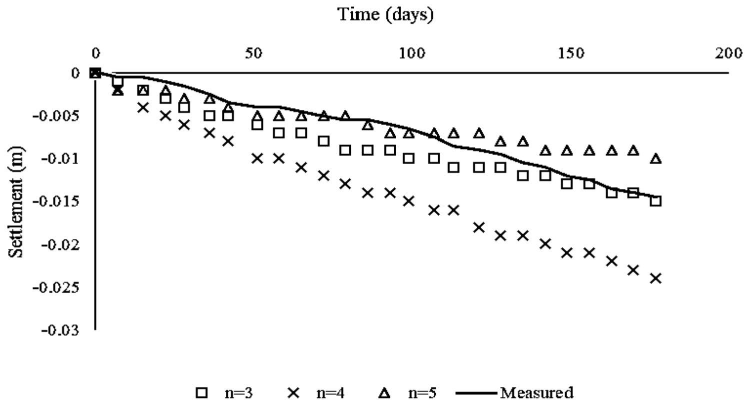


Figure 1. Shows the flowchart of the AI-based models and experimental methods applied



(a)



(b)

Figure 2. The relationship of soil settlement and time, (a) SG1 and (b) SG2

Table 1. The performance of ...

Variable	Speed (rpm)	Power (kW)
x	10	8.6
y	15	12.4
z	20	15.3

3. RESULTS AND DISCUSSION (10 PT)

In this section, it is explained the results of research and at the same time is given the comprehensive discussion. Results can be presented in figures, graphs, tables and others that make the reader understand easily [14], [15]. The discussion can be made in several sub-sections.

3.1. Sub section 1

Equations should be placed at the center of the line and provided consecutively with equation numbers in parentheses flushed to the right margin, as in (1). The use of Microsoft Equation Editor or MathType is preferred.

$$E_v - E = \frac{h}{2.m}\,(k_x^2 + k_y^2) \qquad (1)$$

All symbols that have been used in the equations should be defined in the following text.

3.2. Sub section 2

Proper citation of other works should be made to avoid plagiarism. When referring to a reference item, please use the reference number as in [16] or [17] for multiple references. The use of ”Ref [18]...”

should be employed for any reference citation at the beginning of sentence. For any reference with more than 3 or more authors, only the first author is to be written followed by *et al*. (e.g. in [19]). Examples of reference items of different categories shown in the References section. Each item in the references section should be typed using 9 pt font size [20]–[25].

3.2.1. Subsub section 1

yy

3.2.2. Subsub section 2

zz

4. CONCLUSION (10 PT)

Provide a statement that what is expected, as stated in the "INTRODUCTION" section can ultimately result in "RESULTS AND DISCUSSION" section, so there is compatibility. Moreover, it can also be added the prospect of the development of research results and application prospects of further studies into the next (based on result and discussion).

ACKNOWLEDGEMENTS (10 PT)

Author thanks ... . In most cases, sponsor and financial support acknowledgments.

REFERENCES (10 PT)

The main references are international journals and proceedings. All references should be to the most pertinent, up-to-date sources **and the minimum of references** are **25 entries** (for original research paper) and **50 entries** (for review/survey paper). References are written in **IEEE style**. For more complete guide can be accessed at (http://ipmuonline.com/guide/refstyle.pdf). Use of a tool such as **EndNote**, **Mendeley**, or **Zotero** for reference management and formatting, and choose **IEEE style**. Please use a consistent format for references-see examples (8 pt):

[1] Journal/Periodicals

Basic Format:

J. K. Author, “Title of paper,” *Abbrev. Title of Journal/Periodical*, vol. *x,* no. *x,* pp. *xxx-xxx,* Abbrev. Month, year, doi: *xxx*.

Examples:

- M. M. Chiampi and L. L. Zilberti, “Induction of electric field in human bodies moving near MRI: An efficient BEM computational procedure,” *IEEE Trans. Biomed. Eng.*, vol. 58, pp. 2787–2793, Oct. 2011, doi: 10.1109/TBME.2011.2158315.
- R. Fardel, M. Nagel, F. Nuesch, T. Lippert, and A. Wokaun, “Fabrication of organic light emitting diode pixels by laser-assisted forward transfer,” *Appl. Phys. Lett.*, vol. 91, no. 6, Aug. 2007, Art. no. 061103, doi: 10.1063/1.2759475.

[2] Conference Proceedings

Basic Format:

J. K. Author, “Title of paper,” in *Abbreviated Name of Conf.*, (location of conference is optional), year, pp. *xxx–xxx*, doi: *xxx*.

Examples:

- G. Veruggio, “The EURON roboethics roadmap,” in *Proc. Humanoids ’06: 6th IEEE-RAS Int. Conf. Humanoid Robots*, 2006, pp. 612–617, doi: 10.1109/ICHR.2006.321337.
- J. Zhao, G. Sun, G. H. Loh, and Y. Xie, “Energy-efficient GPU design with reconfigurable in-package graphics memory,” in *Proc. ACM/IEEE Int. Symp. Low Power Electron. Design (ISLPED)*, Jul. 2012, pp. 403–408, doi: 10.1145/2333660.2333752.

[3] Book

Basic Format:

J. K. Author, “Title of chapter in the book,” in *Title of His Published Book*, X. Editor, Ed., *x*th ed. City of Publisher, State (only U.S.), Country: Abbrev. of Publisher, year, ch. *x*, sec. *x*, pp. *xxx–xxx.*

Examples:

- A. Taflove, *Computational Electrodynamics: The Finite-Difference Time-Domain Method* in Computational Electrodynamics II, vol. 3, 2nd ed. Norwood, MA, USA: Artech House, 1996.
- R. L. Myer, “Parametric oscillators and nonlinear materials,” in *Nonlinear Optics*, vol. 4, P. G. Harper and B. S. Wherret, Eds., San Francisco, CA, USA: Academic, 1977, pp. 47–160.

[4] M. Theses (B.S., M.S.) and Dissertations (Ph.D.)

Basic Format:

J. K. Author, “Title of thesis,” M.S. thesis, Abbrev. Dept., Abbrev. Univ., City of Univ., Abbrev. State, year.

J. K. Author, “Title of dissertation,” Ph.D. dissertation, Abbrev. Dept., Abbrev. Univ., City of Univ., Abbrev. State, year.

Examples:

- J. O. Williams, “Narrow-band analyzer,” Ph.D. dissertation, Dept. Elect. Eng., Harvard Univ., Cambridge, MA, USA, 1993.
- N. Kawasaki, “Parametric study of thermal and chemical nonequilibrium nozzle flow,” M.S. thesis, Dept. Electron. Eng., Osaka Univ., Osaka, Japan, 1993.

*In the reference list, however, list all the authors for up to six authors. Use *et al.* only if: 1) The names are not given and 2) List of authors more than 6. *Example*: J. D. Bellamy *et al*., Computer Telephony Integration, New York: Wiley, 2010.

See the examples:

REFERENCES

[1] T. S. Ustun, C. Ozansoy, and A. Zayegh, “Recent developments in microgrids and example cases around the world—A review,” *Renew. Sustain. Energy Rev.*, vol. 15, no. 8, pp. 4030–4041, Oct. 2011, doi: 10.1016/j.rser.2011.07.033.

[2] D. Salomonsson, L. Soder, and A. Sannino, “Protection of Low-Voltage DC Microgrids,” *IEEE Trans. Power Deliv.*, vol. 24, no. 3, pp. 1045–1053, Jul. 2009, doi: 10.1109/TPWRD.2009.2016622.

[3] S. Chakraborty and M. G. Simoes, “Experimental Evaluation of Active Filtering in a Single-Phase High-Frequency AC Microgrid,” *IEEE Trans. Energy Convers.*, vol. 24, no. 3, pp. 673–682, Sep. 2009, doi: 10.1109/TEC.2009.2015998.

[4] S. A. Hosseini, H. A. Abyaneh, S. H. H. Sadeghi, F. Razavi, and A. Nasiri, “An overview of microgrid protection methods and the factors involved,” *Renew. Sustain. Energy Rev.*, vol. 64, pp. 174–186, Oct. 2016, doi: 10.1016/j.rser.2016.05.089.

[5] S. Chen, N. Tai, C. Fan, J. Liu, and S. Hong, “Sequence-component-based current differential protection for transmission lines connected with IIGs,” *IET Gener. Transm. Distrib.*, vol. 12, no. 12, pp. 3086–3096, Jul. 2018, doi: 10.1049/iet-gtd.2017.1507.

[6] S. Parhizi, H. Lotfi, A. Khodaei, and S. Bahramirad, “State of the Art in Research on Microgrids: A Review,” *IEEE Access*, vol. 3, pp. 890–925, 2015, doi: 10.1109/ACCESS.2015.2443119.

[7] S. Chowdhury, S. P. Chowdhury, and P. Crossley, *Microgrids and Active Distribution Networks*. Institution of Engineering and Technology, 2009.

[8] R. Ndou, J. I. Fadiran, S. Chowdhury, and S. P. Chowdhury, “Performance comparison of voltage and frequency based loss of grid protection schemes for microgrids,” in *2013 IEEE Power & Energy Society General Meeting*, 2013, pp. 1–5, doi: 10.1109/PESMG.2013.6672788.

[9] S. Liu, T. Bi, A. Xue, and Q. Yang, “Fault analysis of different kinds of distributed generators,” in *2011 IEEE Power and Energy Society General Meeting*, Jul. 2011, pp. 1–6, doi: 10.1109/PES.2011.6039596.

[10] K. Jennett, F. Coffele, and C. Booth, “Comprehensive and quantitative analysis of protection problems associated with increasing penetration of inverter-interfaced DG,” in *11th IET International Conference on Developments in Power Systems Protection (DPSP 2012)*, 2012, pp. P31–P31, doi: 10.1049/cp.2012.0091.

[11] P. T. Manditereza and R. Bansal, “Renewable distributed generation: The hidden challenges – A review from the protection perspective,” *Renew. Sustain. Energy Rev.*, vol. 58, pp. 1457–1465, May 2016, doi: 10.1016/j.rser.2015.12.276.

[12] D. M. Bui, S.-L. Chen, K.-Y. Lien, Y.-R. Chang, Y.-D. Lee, and J.-L. Jiang, “Investigation on transient behaviours of a uni-grounded low-voltage AC microgrid and evaluation on its available fault protection methods: Review and proposals,” *Renew. Sustain. Energy Rev.*, vol. 75, pp. 1417–1452, Aug. 2017, doi: 10.1016/j.rser.2016.11.134.

[13] T. N. Boutsika and S. A. Papathanassiou, “Short-circuit calculations in networks with distributed generation,” *Electr. Power Syst. Res.*, vol. 78, no. 7, pp. 1181–1191, Jul. 2008, doi: 10.1016/j.epsr.2007.10.003.

[14] H. Margossian, G. Deconinck, and J. Sachau, “Distribution network protection considering grid code requirements for distributed generation,” *IET Gener. Transm. Distrib.*, vol. 9, no. 12, pp. 1377–1381, Sep. 2015, doi: 10.1049/iet-gtd.2014.0987.

[15] O. Núñez-Mata, R. Palma-Behnke, F. Valencia, A. Urrutia-Molina, P. Mendoza-Araya, and G. Jiménez-Estévez, “Coupling an adaptive protection system with an energy management system for microgrids,” *Electr. J.*, vol. 32, no. 10, p. 106675, Dec. 2019, doi: 10.1016/j.tej.2019.106675.

[16] M. Brucoli and T. C. Green, “Fault behaviour in islanded microgrids,” in *Proceedings of the 19th international conference on electricity distribution, CIRED*, 2007, pp. 0548-(1-4).

[17] I. K. Tarsi, A. Sheikholeslami, T. Barforoushi, and S. M. B. Sadati, “Investigating impacts of distributed generation on distribution networks reliability: A mathematical model,” in *Proceedings of the 2010 Electric Power Quality and Supply Reliability Conference*, Jun. 2010, pp. 117–124, doi: 10.1109/PQ.2010.5550010.

[18] L. K. Kumpulainen and K. T. Kauhaniemi, “Analysis of the impact of distributed generation on automatic reclosing,” in *IEEE PES Power Systems Conference and Exposition, 2004.*, pp. 1152–1157, doi: 10.1109/PSCE.2004.1397623.

[19] A. A. Memon and K. Kauhaniemi, “A critical review of AC Microgrid protection issues and available solutions,” *Electr. Power Syst. Res.*, vol. 129, pp. 23–31, Dec. 2015, doi: 10.1016/j.epsr.2015.07.006.

[20] H. A. Abdel-Ghany, A. M. Azmy, N. I. Elkalashy, and E. M. Rashad, “Optimizing DG penetration in distribution networks concerning protection schemes and technical impact,” *Electr. Power Syst. Res.*, vol. 128, pp. 113–122, Nov. 2015, doi: 10.1016/j.epsr.2015.07.005.

[21] S. Chaitusaney and A. Yokoyama, “An Appropriate Distributed Generation Sizing Considering Recloser-Fuse Coordination,” in *2005 IEEE/PES Transmission & Distribution Conference & Exposition: Asia and Pacific*, pp. 1–6, doi: 10.1109/TDC.2005.1546838.

[22] H. H. Zeineldin, Y. A.-R. I. Mohamed, V. Khadkikar, and V. R. Pandi, “A Protection Coordination Index for Evaluating Distributed Generation Impacts on Protection for Meshed Distribution Systems,” *IEEE Trans. Smart Grid*, vol. 4, no. 3, pp. 1523–1532, Sep. 2013, doi: 10.1109/TSG.2013.2263745.

[23] D. Eltigani and S. Masri, “Challenges of integrating renewable energy sources to smart grids: A review,” *Renew. Sustain. Energy Rev.*, vol. 52, pp. 770–780, Dec. 2015, doi: 10.1016/j.rser.2015.07.140.

[24] M. M. Eissa (SIEEE), “Protection techniques with renewable resources and smart grids—A survey,” *Renew. Sustain. Energy Rev.*, vol. 52, pp. 1645–1667, Dec. 2015, doi: 10.1016/j.rser.2015.08.031.

[25] A. Oudalov *et al.*, “Novel Protection Systems for Microgrids,” 2009. [Online]. Available: http://www.microgrids.eu/documents/688.pdf.

BIOGRAPHIES OF AUTHORS (10 PT)
The recommended number of authors is at least 2. One of them as a corresponding author.
Please attach clear photo (3x4 cm) and vita. Example of biographies of authors:

	**Abdel-Rahman Hedar** holds a Doctor of Informatics degree from Kyoto University, Japan in 2004. He also received his B.Sc. and M.Sc. (Mathematics) from Assiut University, Egypt in 1993 and 1997, respectively. He is currently an associate professor at Computer Science Department in Jamoum, Umm Al-Qura University, Makkah, Saudi Arabia. He is also an associate professor of artifcial intelligence in Assiut University since January 2012. His research includes meta-heuristics, global optimization, machine learning, data mining, bioinformatics, graph theory and parallel programming. He has published over 70 papers in international journals and conferences. From July 2005 to July 2007, he was a JSPS research fellow in Kyoto University, Japan. He can be contacted at email: ahahmed@uqu.edu.sa or hedar@aun.edu.eg.
	**Patricia Melin** received the D.Sc. degree (Doctor Habilitatus D.Sc.) in computer science from the Polish Academy of Sciences, Warsaw, Poland, with the Dissertation "Hybrid Intelligent Systems for Pattern Recognition using Soft Computing". She is a Professor of Computer Science in the Graduate Division, Tijuana Institute of Technology, Tijuana, Mexico since 1998. In addition, she is serving as Director of Graduate Studies in computer science and Head of the research group on Computational Intelligence (2000–present). Her research interests are in Type-2 Fuzzy Logic, Modular Neural Networks, Pattern Recognition, Neuro-Fuzzy and Genetic-Fuzzy hybrid approaches., She is currently the President of Hispanic American Fuzzy Systems Association (HAFSA) and is the founding Chair of the Mexican Chapter of the IEEE Computational Intelligence Society. She can be contacted at email: pmelin@tectijuana.mx.
	**Dr. Kennedy Okokpujie** holds a Bachelor of Engineering (B.Eng.) in Electrical and Electronics Engineering, Master of Science (M.Sc.) in Electrical and Electronics Engineering, Master of Engineering (M.Eng.) in Electronics and Telecommunication Engineering and Master of Business Administration (MBA), Ph.D in Information and Communication Engineering, besides several professional certificates and skills. He is currently lecturing with the department of Electrical and Information Engineering at Covenant University, Ota, Ogun State, Nigeria. He is a member of the Nigeria Society of Engineers and the Institute of Electrical and Electronics Engineers (IEEE). His research areas of interest include Biometrics, Artificial Intelligent, and Digital signal Processing. He can be contacted at email: kennedy.okokpujie@covenantuniversity.edu.ng.