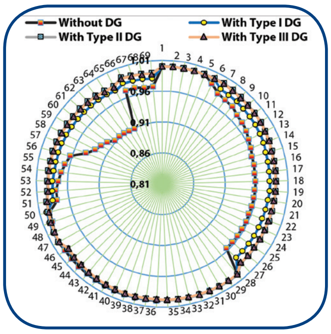
Volume 15, Number 3, 2024

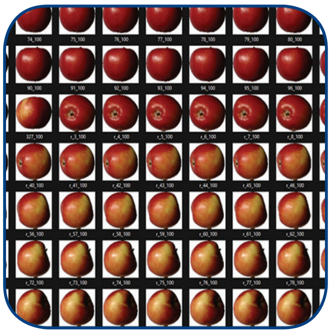
ISSN 1847-6996

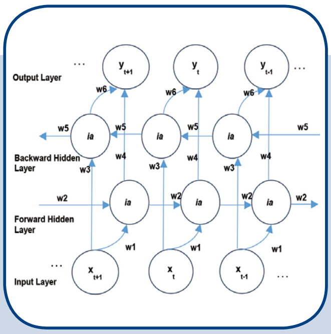
FERIT
FACULTY OF ELECTRICAL ENGINEERING, COMPUTER SCIENCE AND INFORMATION TECHNOLOGY **OSIJEK**

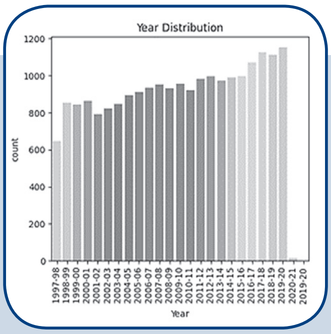
IJECES International Journal of Electrical and Computer Engineering Systems

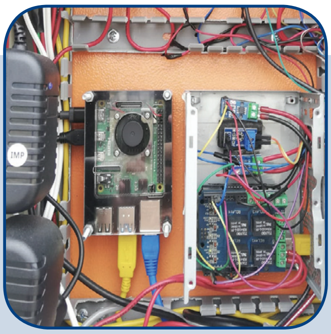
International Journal of Electrical and Computer Engineering Systems











ISSN: 1847-6996

INTERNATIONAL JOURNAL OF ELECTRICAL AND COMPUTER ENGINEERING SYSTEMS

Published by Faculty of Electrical Engineering, Computer Science and Information Technology Osijek,
Josip Juraj Strossmayer University of Osijek, Croatia

Osijek, Croatia | Volume 15, Number 3, 2024 | Pages 215 - 304

The International Journal of Electrical and Computer Engineering Systems is published with the financial support of the Ministry of Science and Education of the Republic of Croatia

CONTACT

International Journal of Electrical and Computer Engineering Systems (IJECES)

Faculty of Electrical Engineering, Computer
Science and Information Technology Osijek,
Josip Juraj Strossmayer University of Osijek, Croatia
Kneza Trpimira 2b, 31000 Osijek, Croatia
Phone: +38531224600, Fax: +38531224605
e-mail: ijeces@ferit.hr

Subscription Information

The annual subscription rate is 50€ for individuals, 25€ for students and 150€ for libraries.
Giro account: 2390001 - 1100016777,
Croatian Postal Bank

Proofreader

Ivanka Ferčec
J.J. Strossmayer University of Osijek, Croatia

Editing and technical assistence

Davor Vrandečić
J.J. Strossmayer University of Osijek, Croatia

Stephen Ward
J.J. Strossmayer University of Osijek, Croatia

Dražen Bajer
J.J. Strossmayer University of Osijek, Croatia

Journal is referred in:

- Scopus
- Web of Science Core Collection (Emerging Sources Citation Index - ESCI)
- Google Scholar
- CiteFactor
- Genamics
- Hrčak
- Ulrichweb
- Reaxys
- Embase
- Engineering Village

Bibliographic Information

Commenced in 2010.
ISSN: 1847-6996
e-ISSN: 1847-7003
Published: quarterly
Circulation: 300

IJECES online

https://ijeces.ferit.hr

TABLE OF CONTENTS

Optimization of Distributed Generation in Radial Distribution Network for Active Power Loss Minimization using Jellyfish Search Optimizer Algorithm

Original Scientific Paper

Jarabala Ranga
Ramachandra College of Engineering
Department of Electrical and Electronics Engineering
Eluru, Andhra Pradesh, India
jarabalaranga@gmail.com

Thiagarajan Y
Christ College of Engineering and Technology
Department of Electrical and Electronics Engineering
Puducherry, India
thiagu2517@gmail.com

Kesavan D
Sona College of Technology
Department of Electrical and Electronics Engineering
Salem, Tamilnadu, India
kesavansalem@gmail.com

Jovin Deglus
Acharya Institute of Technology
Department of Artificial Intelligence and Machine Learning
Bangalore, Karnataka, India
jovin.december@gmail.com

Sibbala Bhargava Reddy
Srinivasa Ramanujan Institute of Technology
Department of Electrical and Electronics Engineering
Anantapur, Andhra Pradesh, India
bhargav.s204@gmail.com

Rajakumar P
Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology
Department of Electrical and Electronics Engineering
Avadi, 400 Feet Outer Ring Road, Chennai, Tamilnadu, India
drrajakumarp@veltech.edu.in

Priya. R. A
Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology
Department of Electrical and Electronics Engineering
Avadi, 400 Feet Outer Ring Road, Chennai, Tamilnadu, India
drpriyara@veltech.edu.in

Abstract *– The inclusion of distributed generation (DG) units in the distribution network (DN) effectively cuts down the power losses (PL) and strengthens the voltage profile (VP). This paper examines the effect of allocating different distributed generation (DG) in radial distribution networks (RDN) through an implementation of an optimization technique using a recently introduced bio-inspired algorithm known as a jellyfish search optimizer (JSO). Unlike the other optimization algorithms, the JSO algorithm evades the local optimal trap and reaches the optimal solution in less time. The DG position(s) and size(s) are optimized for active power loss (APL) minimization with respect to several constraints. The effectiveness and robustness of the proposed optimization technique using JSO algorithm is investigated on a balanced IEEE RDNs with 33, 69 and 118-buses. The simulation outcomes are obtained for different types (type I, II and III) of DG placement. Additionally, a comprehensive comparative study has been performed for the JSO and other algorithms. The comparison exemplifies that the proposed JSO optimization approach produces a better optimal solution with steady convergence than other techniques reported in the literature. Also, the simulation findings show the potentiality of JSO optimization method for solving complex optimization problems.*

Keywords*:* *Distributed Generation, Radial Distribution Network, Jellyfish Search Optimizer, Active Power Losses*

Received: October 12, 2023; Received in revised form: January 16, 2024; Accepted: February 13, 2024

1. INTRODUCTION

The electrical power system generates electricity and transfers it to consumers through conductors via transmission and distribution systems. The transmission system (TS) carries the electricity from the generating plant to the distribution system (DS) using high-tension conductors. Then, from 'DS' the power is delivered to consumers through low-tension conductors via distribution power networks (DPN). In the process of power transfer, a portion of energy is lost as power losses in transmission and distribution systems. Literature reports [1] that about 70% of power losses (PL) occur in DS and the remaining 30% in TS. The PL in DS is more than TS because of its radial structural design, a higher line R/X ratio and a greater number of load buses [2]. However, an efficient, secure and reliable DPN should account a less PL and voltage drop.

In recent times, a unique power generation technology known as distributed generation (DG) is introduced in DS to achieve numerous technical, economic and ecological benefits including PL minimization (both active and reactive losses), voltage profile (VP) improvement, stability enhancement, operating cost reduction and greenhouse gas emission minimization. DG injects electrical power at/near load points [3]. However, the utility gets benefit through DG placement only when its location and size are optimized in DPN. Numerous optimization techniques implemented by researchers over the years to assimilate DG unit(s) optimally into DPN. A few of them are outlined below.

An analytical methodology [4, 5] and iterative methodology [6] proposed to minimize PL in DPN. A modified aquila optimizer (MAO) technique introduced [7] to reduce the APL and to improve the VP of radial distribution network (RDN) for different types of renewable DG placement. The proposed technique was tested on IEEE 33-bus RDN. The authors [8] implemented an optimization approach using an improved wild horse optimization (IWHO) algorithm to optimize DG units in IEEE RDNs 33, 69 and 119-buses for the APL minimization. A novel hybrid optimization approach proposed [9] combining simulated annealing (SA) and particle swarm optimization (PSO) algorithms to optimize DG position and size into RDN for the APL minimization. The simulation study was executed for IEEE 33-bus RDN. The authors [10] applied a rider optimization algorithm (ROA) for locating the optimal site and computing optimal size for the different renewable energy sources (PV, WT and biomass) in RDN. The proposed ROA approach optimized the DG for minimizing total APL. A novel DG optimization technique proposed [11] using shark optimization algorithm (SOA) to minimize the PL, to enrich the voltage profile and voltage stability of RDN. A hybrid technique based on LSF and SA proposed [12] to optimize PV and WT in IEEE 33 and 69-bus RDNs for APL minimization and voltage enhancement. The authors [13] have implemented an improved version of symbiotic organisms search (SOS) algorithm known as the quasi-oppositional chaotic SOS algorithm to optimally incorporate DGs with different power factor (p.f) into IEEE 33, 69 and 118-buses for the benefit of PL reduction, VP improvement and voltage stability enhancement. An optimization approach using chaotic sine cosine algorithm (CSCA) proposed [14] to optimize multi-DG units into IEEE 33 and 69-bus RDN for solving a single and multiple objectives DG allocation problem. A harris hawks optimization (HHO) algorithm applied [15] to solve single and multi-DG placement problems in RDN. The DGs with different p.f optimized into 33 and 69-buses RDN to reduce PL, enrich VP and improve stability. A new hybrid approach proposed [16] using improved GWO and PSO to optimize DG location and size in RDN to achieve PL reduction, VP enrichment and voltage stability enhancement. The proposed approach adopted a dimension learning hunting method to optimize the DG. Genetic algorithm (GA) and PSO algorithm were proposed [17] to optimally assimilate single and multiple (two) PV and WT DGs into 33-bus RDN. GWO algorithm based DG planning executed [18] for RDN to cut down the PL. The proposed approach optimized the different DGs into IEEE standard test systems with 16, 30, 57 and 118-buses. Water cycle algorithm (WCA) based optimization technique implemented [19] to optimize multi-DGs (FC, PV and WT) into RDN for minimization of total APL, operating cost and greenhouse gas discharge. The authors [20] proposed an integrated technique using LSF and sine cosine algorithm (SCA) to optimize PV and WT for the objectives of PL reduction and VP improvement. The proposed method executed on unbalanced IEEE RDN with 33 and 69-buses.

Above-mentioned techniques have been implemented for solving DG placement problems in RDN and provided reasonable solutions. However, literature [4-6] reported that the analytical techniques suffer from inadequate solutions and convergence problems. Likewise, most of the optimization algorithms offer a chance for premature convergence and produce local optimal solutions. In recent times, many novel algorithms are introduced to solve various complex optimization problems. One such algorithm is known as Jellyfish Search Optimizer (JSO) [21]. The JSO is a swarm-based algorithm that simulates the food-searching manners of jellyfish to produce optimal solutions for a given problem. The JSO has the ability to converge faster than the other algorithms using its stronger searching technique. Also, the JSO requires only few parameter initializations and exhibits better balance between exploration and exploitation. Furthermore, the JSO performance has been tested with numerous benchmark functions and has provided a near optimal solution at rapid convergence [21]. The contribution of the proposed research work is summarized below.

- Propose a new optimization technique using JSO algorithm to optimize the different DG (type I, II and III) units in RDN for APL reduction.
- Apply the proposed JSO algorithm to identify the optimal site (s) and size(s) for different DG types to minimize total APL of RDN.
- Investigate the robustness of the proposed methodology for small (33-bus), medium (69-bus) and large (118-bus) RDN. And, validate the JSO optimized research findings through a comprehensive comparison.

The remaining portion of the manuscript is structured with different sections as follows: Section 2 presents the objective function framework and necessary constraints. Section 3 details the concept and mathematical modelling of the JSO algorithm. Section 4 presents the simulation findings for the IEEE 33-bus, 69-bus and 118-bus RDN for different DG placement and Section 5 highlights the simulation outcome of the JSO technique as a conclusion.

2. PROBLEM FORMATION

The optimal site(s) and size(s) for the *DG* unit(s) are optimized for an objective of minimizing the total *APL* of *RDN*. The total active power loss (APL_T) in a *RDN* represented in Fig. 1 is calculated using Eq. 1.

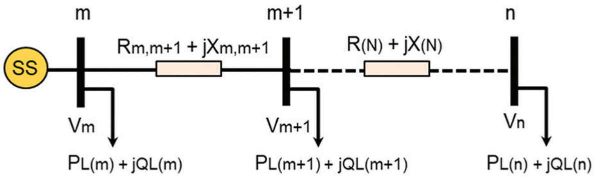


Fig. 1. Radial distribution network

$$APL_T = \sum_{m=1}^{n-1} R_{m,m+1} * \left(\frac{P_m{}^2 + Q_m{}^2}{|V_m|^2} \right) \quad (1)$$

Where, *R* corresponds to distribution line resistance; P_L & Q_L refer active and reactive power demand, respectively, *m* and *n* are buses and *V* is a voltage of buses.

The fitness function or objective function for *APL* minimization is expressed as given in Eq. 2.

$$F = \min \left(\frac{APL_{T(\text{after DG})}}{APL_{T(\text{before DG})}} \right) \quad (2)$$

2.1. CONSTRAINTS

The *DG* sizes are optimized to minimize the APL_T according to several operating parameter constraints of *RDN* including voltage magnitude, feeder current and power flow.

Bus voltage constraint:

$$0.95\text{p.u} \leq V_m \leq 1.05\text{p.u} \quad (3)$$

Thermal constraint:

$$I_{m,m+1} \leq I_{m,m+1}^{\max} \quad (4)$$

***DG* active power (*PDG*) injection constraint:**

$$P_{DG}^{\min} \leq P_{DG} \leq P_{DG}^{\max} \quad (5)$$

***DG* reactive power (Q_{DG}) injection constraint:**

$$Q_{DG}^{\min} \leq Q_{DG} \leq Q_{DG}^{\max} \quad (6)$$

Power balance constraint:

$$P_{swing} + \sum_{i=1}^{N_{DG}} P_{DG}(i) = \sum_{m=1}^{n} P_m + \sum_{m=1}^{N} P_{loss,m} \quad (7)$$

$$Q_{swing} + \sum_{i=1}^{N_{DG}} Q_{DG}(i) = \sum_{m=1}^{n} Q_m + \sum_{m=1}^{N} Q_{loss,m} \quad (8)$$

Where, '*I*' is the magnitude of branch current; P_{DG}, $P_{DG\text{-}min}$ and $P_{DG}{}^{max}$ are the optimal, minimum and maximum real power capacity of *DG* unit, respectively; Q_{DG}, $Q_{DG}{}^{min}$ and $Q_{DG}{}^{max}$ are the optimal, minimum and maximum reactive power capacity of *DG* unit, respectively; *n* and *N* refer to a total number of buses and branches in *RDN*, respectively.

The power flow (*PF*) analysis in *DPN* is important for assessing the various parameters including power losses and bus voltages. The power flow methods suitable for transmission power networks such as Gauss-Seidel and Newton Raphson algorithms have become inappropriate for *RDPN* due to its unique radial structure and higher line *R/X* value. Hence, for an accurate and optimal power flow solution, *RDPN* implements *PF* study using the backward/forward sweep (*BFS*) algorithm [9]. In this study, *BFS* algorithm is executed for *PF* study.

3. SOLUTION METHODOLOGY: JELLYFISH SEARCH OPTIMIZER ALGORITHM

Jellyfish search optimizer (JSO) [21] is a recent algorithm inducted into the group of metaheuristic algorithms for solving an optimization problem. JSO is a swarm-based algorithm and it makes use of the food searching process of jellyfish. The jellyfish search food (fish eggs, larvae, etc.,) stochastically in the ocean. The jellyfish follow two types of search movement: (i) Ocean current (OC) and (ii) Jellyfish swarm [21]. The JSO incorporate two phases of search technique such as diversification and intensification. It also has a time control mechanism to switch between these two search phases. The mathematical modelling of different phases of the JSO algorithm is discussed in the subsequent subsections.

3.1. POPULATION INITIALIZATION

The JSO adopt a unique approach called chaotic map [21] to initialize the population size rather than a typical random process. This effectively eliminates the probability of local optima stagnation and premature convergence as in the case of random process initialization. Equation 10 expressed the population initialization in JSO.

$$X_{i+1} = \eta X_i (1 - X_i), \quad 0 \leq X_i \leq 1 \quad (10)$$

Where, *X* refers to the logistic chaotic value of jellyfish, X_0∈("0,1") X_0∈{"0,0.25,0.75,0.5,1.0" } and *η* is a constant.

3.2. FOLLOWING OCEAN CURRENT

The OC has rich quantities of nutrients and the jellyfish follows OC to search food. The aggregation of all the vectors from populated jellyfish to best (current) jellyfish is used to determine the direction of OC ($\overrightarrow{trend}$). Equation 11 simulates the OC direction [21].

$$\overrightarrow{\text{trend}} = X^* - \beta \times \text{rand}(0,1) \times \mu \quad (11)$$

Where, *X** points to the best position of jellyfish (current best); *β* and *μ* refer to distribution coefficient concerning to the length of ($\overrightarrow{\text{trend}}$) and mean position of all jellyfish, respectively. Typically, *β* value is more than zero. Consequently, the position of each jellyfish is updated using Eq. 12 and Eq. 13.

$$X_i(t+1) = X_i(t) + \text{rand}(0,1) \times \overrightarrow{\text{trend}} \quad (12)$$

$$X_i(t+1) = X_i(t) + \text{rand}(0,1) \times X^* - \beta \times \text{rand}(0,1) \times \mu \quad (13)$$

3.3. JELLYFISH SWARM

The jellyfish move around the swarm in two motions [21]: passive and active. The passive and active movements of the jellyfish are termed as type A and type B motions, respectively. During the early stages of the swarm formation, the majority of jellyfish follow the type 'A' motion and after a period of time they tend to follow the type 'B' motion. The type 'A' motion refers to the movement of jellyfish around its own position. The updated position of jellyfish after type 'A' motion is given by Eq. (14).

$$X_i(t+1) = X_i(t) + \gamma \times rand\ (0,1) \times \lfloor Ub - Lb \rfloor \quad (14)$$

Where, L_b and U_b correspond to the lower and upper search space limit, respectively; *γ* is a motion coefficient and depends upon the length of motion around individual jellyfish.

The direction of jellyfish movement in type 'B' motion is determined by considering a jellyfish (*j*) beside the one selected in the random process and a vector from ith jellyfish to jth jellyfish. The jellyfish (*i*) will move towards the direction of jellyfish (*j*) when the quantity of food available in jellyfish (*j*) is more than the position of jellyfish (*i*). However, the jellyfish (*i*) moves away from jellyfish (*j*) if food availability at the position of jellyfish (*j*) is lower than jellyfish (*i*). Likewise, all jellyfish move around the swarm to locate a better position for finding food. The mathematical representation for jellyfish motion and its position updation is given in Eq. 15, Eq.16 and Eq.17.

$$\overrightarrow{\text{Step}} = \text{rand}(0,1) \times \overrightarrow{\text{Direction}} \quad (15)$$

$$\overrightarrow{\text{Direction}} = \begin{cases} X_j(t) - X_i(t) & \text{if} \quad f(X_i) \geq f(X_j) \\ X_i(t) - X_j(t) & \text{if} \quad f(X_i) < f(X_j) \end{cases} \quad (16)$$

Hence

$$X_i(t+1) = X_j(t) + \overrightarrow{\text{Step}} \quad (17)$$

Where, *f* is an objective function of location *X*.

3.4. TIME CONTROL MECHANISM

The jellyfish forms a swarm and search food in the ocean current (*OC*). The *OC* changes, whenever the temperature or the wind direction changes. Under this circumstance, the jellyfish creates one more swarm and moves toward another *OC*. This motion of jellyfish within the swarms can be categorized into a type 'A' and type 'B' motion where a jellyfish typically moves or switches position. A jellyfish follows type 'A' motion especially at the beginning of the hunt and after a while, it gets favor from type 'B' motion. In order to simulate this switchover mechanism, a time control technique is introduced in the JSO algorithm. A time control function (*TCF*), *c* (*t*) and a constant, c_0 are introduced to regulate the movement of a jellyfish between *OC* and swarm. The *TCF* is a random number between 0 and 1 which fluctuates over time. The mathematical illustration of *TCF* is given in Eq.18. The jellyfish move towards *C* when the *TCF* value is more than c_0. But, a jellyfish move within the swarm if TCF is less than c_0. The value of c_0 is unknown and it will vary randomly between 0 and 1. However, the c_0 value is taken as 0.5, taking the average values of 0 and 1.

$$c(t) = \left| \left(1 - \frac{t}{\text{Iter}_{\max}}\right) \times (2 \times \text{rand}(0,1) \ - 1) \right| \quad (18)$$

Where, *t* and $Iter_{max}$ correspond to iteration time and a maximum number of iterations, respectively. The expression (1- *c* (*t*)) represents the motion of jellyfish inside a swarm. The jellyfish follows type 'A' motion when rand (0, 1) exceeds (1- *c* (*t*)), if not then the jellyfish follows type 'B' motion. Initially, the probability of rand (0, 1) > (1- *c* (*t*)) is higher than later. Hence, jellyfish prefer type 'A' motion at the beginning of the search and then switch over to type 'B' motion after a while.

3.5. BOUNDARY CONDITIONS

The movement of jellyfish inside an ocean is a random process and its position should be normalized whenever it violates the boundary condition for better performance. Equation 19 illustrates the random process and boundary condition.

$$X'_{i,d} = \begin{cases} (X_{i,d} - U_{b,d}) + L_{b,d} & \text{if} \quad X_{i,d} > U_{b,d} \\ (X_{i,d} - L_{b,d}) + U_{b,d} & \text{if} \quad X_{i,d} < L_{b,d} \end{cases} \quad (19)$$

Where, $X_{i,d}$ and $X_{i,d}'$ denote i^{th} jellyfish's actual position and updated position after boundary normalization, respectively. $L_{b,d}$ and $U_{b,d}$ are the lower and upper boundary conditions of the search area, respectively. Fig. 2 illustrates the flowchart of the JSO algorithm.

4. TEST RESULTS AND DISCUSSION

This section presents the simulation findings of JSO algorithm optimized DG units for IEEE standard 33-bus, 69-bus and 119-bus RDNs. The necessary codes of programming are executed in MATLAB software version 2020b. The simulation study was executed 30 times for 100 iterations. The control parameter and the necessary constraints for the JSO algorithm is presented in Table 1.

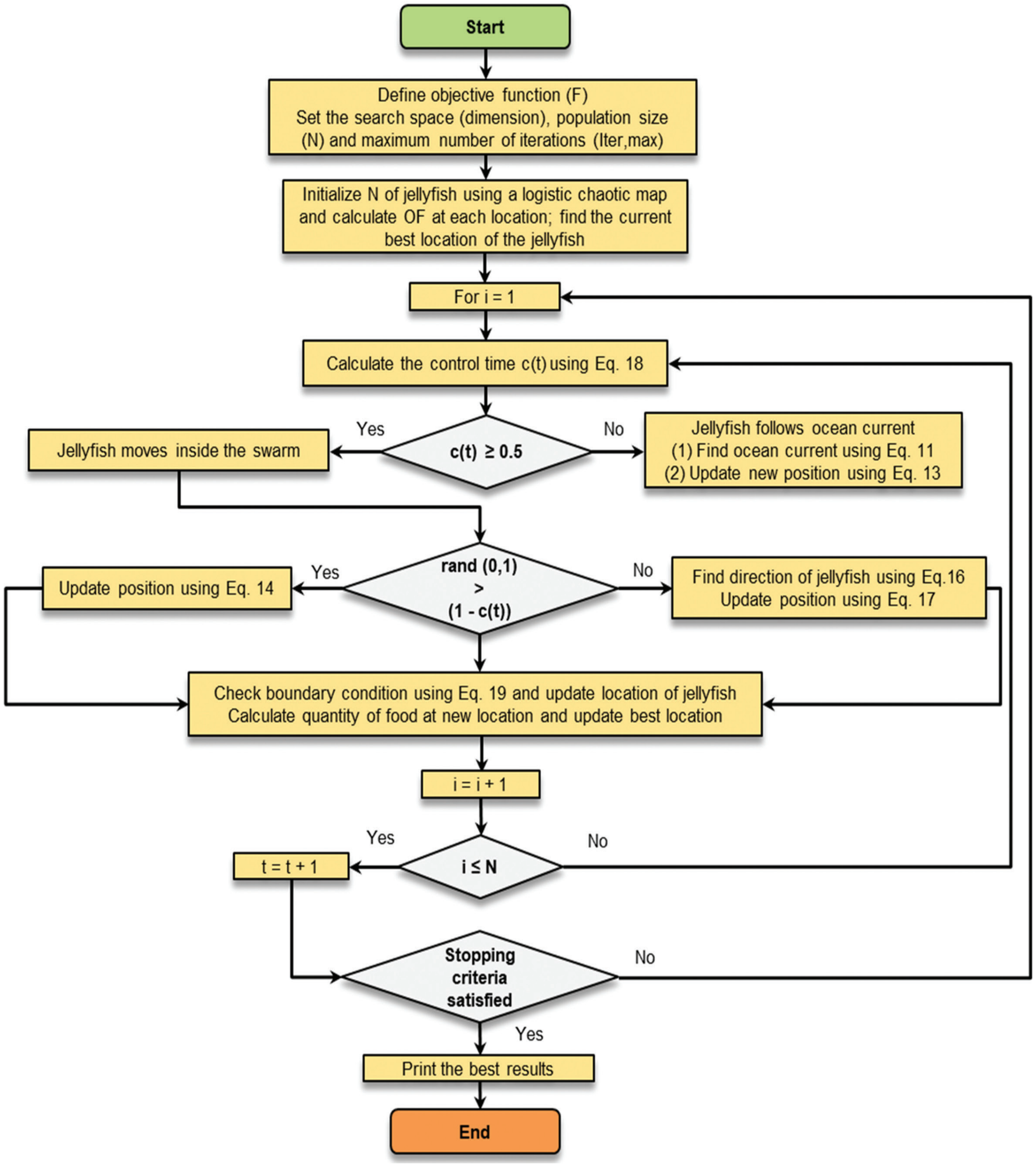

Start
Define objective function (F)
Set the search space (dimension), population size (N) and maximum number of iterations (Iter,max)
Initialize N of jellyfish using a logistic chaotic map and calculate OF at each location; find the current best location of the jellyfish
For i = 1
Calculate the control time c(t) using Eq. 18
Yes
Jellyfish moves inside the swarm
c(t) ≥ 0.5
No
Jellyfish follows ocean current
(1) Find ocean current using Eq. 11
(2) Update new position using Eq. 13
Yes
Update position using Eq. 14
rand (0,1) > (1 - c(t))
No
Find direction of jellyfish using Eq.16
Update position using Eq. 17
Check boundary condition using Eq. 19 and update location of jellyfish
Calculate quantity of food at new location and update best location
i = i + 1
Yes
t = t + 1
i ≤ N
No
Stopping criteria satisfied
No
Yes
Print the best results
End

Fig. 2. Flowchart of JSO algorithm

Table 1. Control parameter and constraints

Variable	Values
No. of populations	30
No. of iterations	100
Base MVA	100
Bus voltage constraint	0.95 p.u<V_i<1.05 p.u
DG capacity limit	33-bus *RDN* - 400<P_{DG}<3000 250<*QDG*<1830
	69-bus *RDN* - 400<P_{DG}<3100 300<*QDG*<2100
	118-bus *RDN* - 2400<P_{DG}<18000 1750<*QDG*<13250

The simulation study is executed considering the following assumptions.

- The *RDN* power demand is constant and balanced.
- The environmental climate irregularity for *DG* modelling is ignored.
- The *PF* results of *RDN* without *DG* accommodation are referred as base case results.

The proposed simulation study has been executed to optimize the location and size for type I (photovoltaic), II (capacitors) and III (synchronous generator) *DG* to minimize total *APL* of *RDN*.

The following subsections present the simulation findings of different *RDN*s with and without *DG* accommodation.

4.3 SIMULATION OUTCOME WITH NO DG PLACEMENT

The simulation outcome for different IEEE *RDN*s with no *DG* placement is presented in Table 2. The *PF* execution using *BFS* algorithm for 33-bus, 69-bus and 118-bus *RDN*s without *DG* results in 210.98 kW, 225 kW and 1296.3 kW total *APL*, respectively. Noticeably, 21 out of 33-bus, 9 out of 69-bus and 45 out of 118-bus *RDN*s violates the minimum bus voltage (V_{min}) constraint and register V_{min} 0.9038 p.u, 0.9092 p.u and 0.8688 p.u, respectively.

Table 2. Simulation outcome: Without *DG* accommodation

Outcome	IEEE 33-bus RDN	IEEE 69-bus RDN	IEEE 118-bus RDN
Active power demand in MW	3.72	3.8	22.71
Reactive power demand in MVAr	2.3	2.69	17.04
Total APL in kW	210.98	225	1296.3
Vmin in p.u.	0.9038	0.9092	0.8688

4.3 SIMULATION OUTCOME WITH DG PLACEMENT

The simulation findings for different *RDN*s with *DG* units are presented in Table 3.

Table 3. Simulation outcome: With *DG* accommodation

Outcome	IEEE 33-bus RDN			IEEE 69-bus RDN			IEEE 118-bus RDN		
	DG Type			*DG* Type			*DG* Type		
	I (kW)	II (kVAr)	III (kVA)	I (kW)	II (kVAr)	III (kVA)	I (kW)	II (kVAr)	III (kVA)
Location	30	30	30	61	61	61	61,17,65,12,13	61,17,65,12,13	61,17,65,12,13
Size	2133.67	1647.12	2689.43	1798.65	1328.25	1957.54	2660.1,1796.5, 2353.2,1636.7, 1786.9	1850.1,1923.5, 2003.2,1696.7, 2326.9	2650.1,1995.5, 2103.2,1896.7, 2006.9
Total *APL* in kW	101.8	146.1	60.46	71.24	133.24	20.38	456.78	678.23	187.46
V_{min} in p.u.	0.9522	0.9512	0.965	0.9776	0.9855	0.9845	0.9785	0.9932	0.9894

4.3.1. IEEE 33-bus RDN:

Graphical illustrations for APL and VP of IEEE 33-bus RDN before and after DG deployment are presented in Fig. 3 and Fig.4, respectively. The optimal allocation of DGs results in significant power loss reduction. The total APL of the test network has reduced to 101.8 kW, 146.1 kW and 60.46 kW respectively for type I, II and III optimized DG allocation. Also, the Vmin of the 33-bus RDN enhanced to 0.9522p.u, 0.9512p.u and 0.965p.u after the addition of type I, II and III DG, respectively and no buses of the power network fall below 0.95p.u.

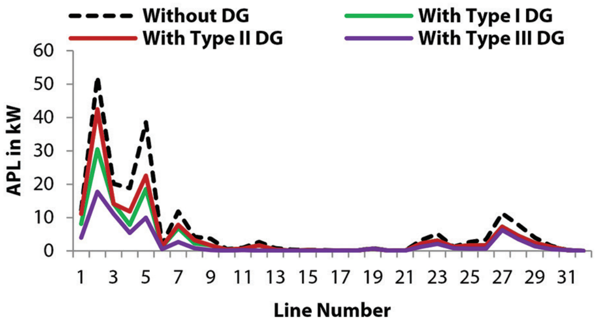


Fig. 3. APL of IEEE 33-bus RDS prior and after DG allocation

Moreover, JSO optimized DG placement converges to optimal result taking 7, 9 and 12 iterations and consuming 12, 14 and 17 seconds of CPU time for type I, II and III DG respectively. Fig. 5 shows the convergence characteristic of JSO algorithm for 33-bus RDN.

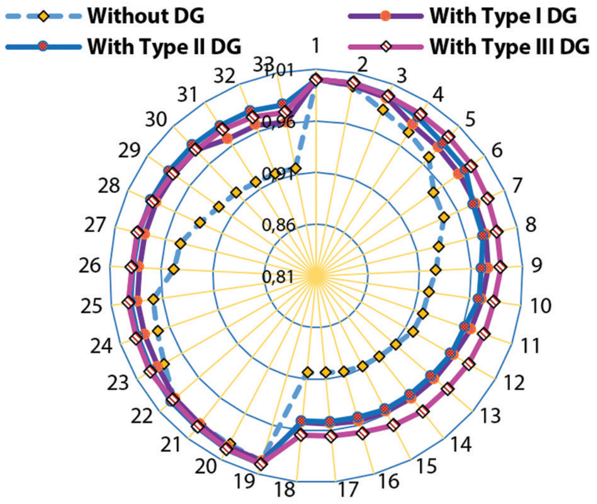


Fig. 4. VP of IEEE 33-bus RDS prior and after DG allocation

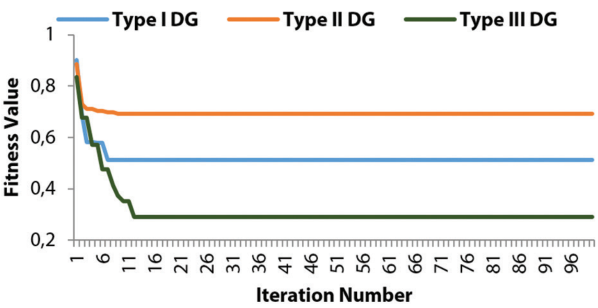


Fig. 5. Convergence curve of JSO algorithm for 33 bus RDS

4.3.2. IEEE 69-bus RDN:

PF execution for the 69-bus RDN with optimal type I, II and III DG placement results in a total APL of 71.24 kW, 133.24 kW and 20.38 kW, respectively. Furthermore, the Vmin of the test network increased to 0.9776p.u, 0.9855p.u and 0.9845p.u for type I, II and III DG respectively. Figs. 6 and 7 illustrate the APL and VP of 69-bus RDN prior and after DG allocation, respectively.

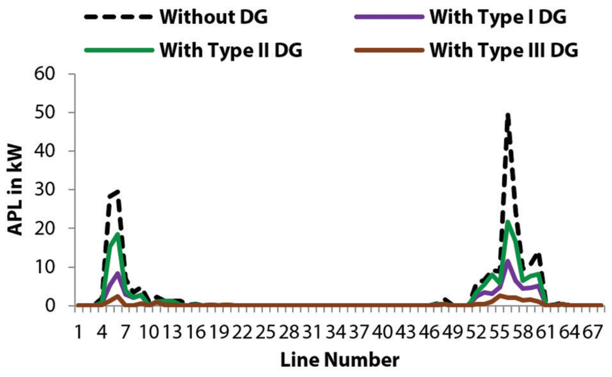


Fig. 6. APL of IEEE 69-bus RDS prior and after DG allocation

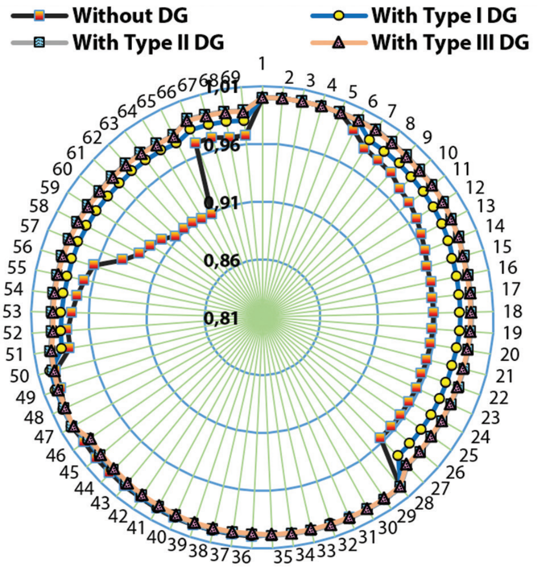


Fig. 7. VP of IEEE 69-bus RDS prior and after DG allocation

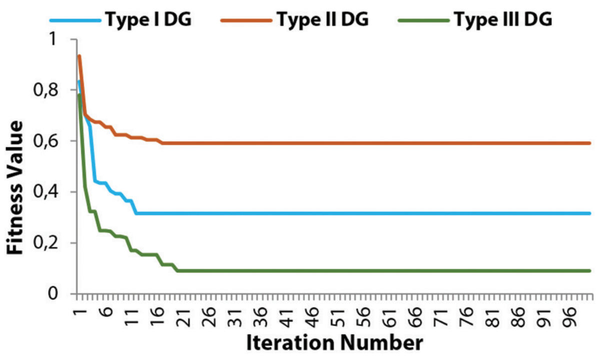


Fig. 8. Convergence curve of JSO algorithm for 69-bus RDS

The JSO converges to optimal result taking 11, 17 and 20 iterations and consumes 15, 25 and 31 seconds of CPU time respectively for type I, II and III optimized DG placement. The convergence characteristic of the JSO algorithm for 69-bus RDN is shown in Fig. 8.

4.3.3. IEEE 118-bus RDN:

The robustness of the JSO algorithm is examined by extending the simulation study to a large and complex 118-bus RDN. The number DGs for optimization are increased to five considering a large RDN. Table 3 presents optimal locations and the corresponding sizes of multi-DG for IEEE 118-bus RDN. The PF execution of a test network after multiple type I, II and III DG allocation minimized the total APL to 456.78 kW, 678.23 kW and 187.46 kW, respectively. The DG allocation also enriched the Vmin of the test network significantly to 0.9785p.u for type I, 0.9932p.u for type II and 0.9894p.u for type III. The VP of 118-bus RDN prior and after the allocation of multiple DGs is presented in Fig. 9.

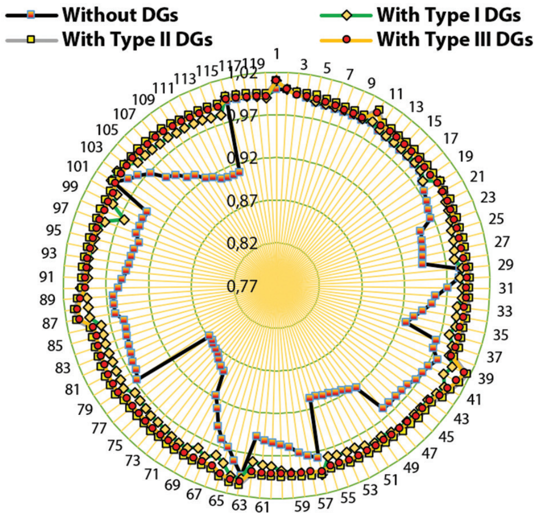


Fig. 9. VP of IEEE 118-bus RDS prior and after DG allocation

The proposed optimization technique converges to optimal solution taking 37, 36 and 41 iterations and consumes 52, 49 and 63 seconds of CPU time respectively for multiples of type I, II and III DG placement. Fig. 10 shows the convergence curve of the JSO algorithm for 118-bus RDN.

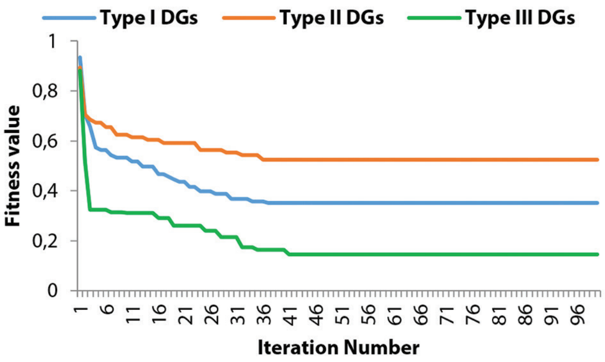


Fig. 10. Convergence curve of JSO algorithm for 118-bus RDS

The simulation findings presented in Table 3 also highlight that Type III DG deployment results more power loss reduction than type I and II DGs by injecting both active (P) and reactive (Q) powers into RDN.

4.4. COMPARATIVE ANALYSIS

In order to showcase the supremacy of the JSO algorithm, the simulation findings of the JSO algorithm are compared with the other algorithms cited in the literature. Table 4 presents the comparative results for type I and type III DG placement in 33-bus RDN. A comparison has revealed that the proposed JSO algorithm outclassed other optimization algorithms (SOA [11], ROA [10], SCA [20], HHO [15], LSF-SA [12] and WHO [8]) delivering a higher percentage of APL reduction at reduced DG capacity. Furthermore, JSO algorithm seamlessly converges to the best solution without trapping in local optima solution and tool less no. of iteration for convergence.

Table 4. Comparison result: IEEE 33-bus RDN with type I and III DG

Parameter	Type I DG				Type III DG					
	SOA [11]	ROA [10]	SCA [20]	Proposed	SOA [11]	ROA [10]	HHO [15]	LSF-SA [12]	WHO [8]	Proposed
Location	6	6	6	30	6	6	26	6	6	30
Size	2600	2590.2	2590.1	2133.67	2550	3144.6	2952.95	3098.2	3081.7	2689.43
Total APL in kW	102.8	111.02	111.02	101.8	65.1426	67.83	69.443	67.8118	61.3147	60.46
No. of iterations	NR	NR	NR	7	NR	17	28	28	15	12
CPU time (sec)	NR	NR	NR	12	NR	NR	NR	NR	NR	17

5. CONCLUSION

In this work, a novel optimization technique has been introduced using a jellyfish search optimizer (JSO) algorithm to optimize DG into RDN to minimize total APL. The optimal site and size for different DG (type I, II and III) were optimized using the JSO algorithm. The simulation study has been implemented on IEEE 33, 69 and 118-bus RDNs for different DG allocation. JSO optimized type I, II and III DG allocation in IEEE 33-bus RDN result 51.74%, 30.75% and 71.34% of total APL, respectively. For IEEE 69-bus RDN, type I, II and III DG placement reduced the total APL by 68.33%, 40.78% and 90.94%, respectively. Likewise, for multiple allocation of type I, II and III DGs in 118-bus RDN cut down the APL by 64.76%, 47.67% and 85.53%, respectively. In addition, the optimized solution enhanced the voltage profile of the RDNs significantly above the specified level (0.95p.u). The simulation finding of JSO for optimized DG allocation emphasizes its ability to find better solutions for complex optimization problems.

6. REFERENCES

[1] M. Ali, M. Mohammed, O. Mohammed, “Optimal Network Reconfiguration Incorporating with Renewable Energy Sources in Radial Distribution Networks”, International Journal of Advanced Science and Technology, Vol. 29, No. 12, 2020, pp. 3114-3133.

[2] P. Chiradeja, R. Ramkumar, “An Approach to Quantify the Technical Benefits of Distribute Generation” IEEE Transactions on Energy Conversion, Vol. 19, No. 4, 2004, pp. 1686-1693.

[3] A. Naresh, M. Pukar, N. Mithulananthan, “An Analytical Approach for DG Allocation in Primary Distribution Network”, International Journal of Electrical Power and Energy Systems, Vol. 28, No. 10, 2006, pp. 669-678.

[4] S. Ghosh, S. P. Ghoshal, S. Ghosh, “Optimal Sizing and Placement of Distributed Generation in a Network System”, International Journal of Electrical Power and Energy Systems, Vol. 32, No. 8, 2010, pp. 849-54.

[5] S. G. Naik, D. Khatod, M. Sharma, “Optimal Allocation of Combined DG and Capacitor for Real Power Loss Minimization in Distribution Networks”, International Journal of Electrical Power and Energy Systems, Vol. 53, 2013, pp. 967-973.

[6] A. Bayat, A. Bagheri, “Optimal Active and Reactive Power Allocation in Distribution Networks using a Novel Heuristic Approach”, Applied Energy, Vol. 233, 2019, pp. 71-85.

[7] M. H. Ali, A. T. Salawudeen, S. Kamel, H. B. Salau, M. Habil, M. Shouran, “Single- and Multi-Objective Modified Aquila Optimizer for Optimal Multiple Renewable Energy Resources in Distribution Network”, Mathematics, Vol. 10, 2022, p. 2129.

[8] M. H. Ali, S. Kamel, M. H. Hassan, M. Tostado-Véliz, H. M. Zawbaa “An Improved Wild Horse Optimization Algorithm for Reliability Based Optimal DG Planning of Radial Distribution Networks”, Energy Reports, Vol. 8, 2022, pp. 582-604.

[9] M. H. Ali, M. Mehanna, E. Othman “Optimal Planning of RDGs in Electrical Distribution Networks

using Hybrid SAPSO Algorithm", International Journal of Electrical and Computer Engineering, Vol. 10, No. 6, 2020, pp. 6153-6163.

[10] M. Khasanov, S. Kamel, C. Rahmann, H. M. Hasanien, A. Al-Durra, "Optimal Distributed Generation and Battery Energy Storage Units Integration in Distribution Systems Considering Power Generation Uncertainty", IET Generation. Transmission and Distribution, Vol. 15, 2021, pp. 3400-3422.

[11] E. S. Ali, S. M. Abd Elazim, S. H. Hakmi, M. I. Mosaad, "Optimal Allocation and Size of Renewable Energy Sources as Distributed Generations Using Shark Optimization Algorithm in Radial Distribution Systems", Energies, Vol. 16, 2023, p. 3983.

[12] S. A. Nowdeh, I. F. Davoudkhani, M. J. H. Moghaddam, E. S. Najmi, A. Y. Abdelaziz, A. Ahmadi, "Fuzzy Multi-Objective Placement of Renewable Energy Sources in Distribution System with Objective of Loss Reduction and Reliability Improvement Using a Novel Hybrid Method", Applied Soft Computing Journal, Vol. 77, 2019, pp. 761-779.

[13] K. H. Truong, P. Nallagownden, I. Elamvazuthi, "A Quasi-Oppositional-Chaotic Symbiotic Organisms Search Algorithm for Optimal Allocation of DG in Radial Distribution Networks", Applied Soft Computing Journal, Vol. 88, 2020, p. 106067.

[14] A. Selim, S. Kamel, F. Jurado, "Efficient Optimization Technique for Multiple DG Allocation in Distribution Networks", Applied Soft Computing Journal,Vol. 86, 2020, p. 105938.

[15] A. Selim, S. Kamel, A. S. Alghamdi, F. Jurado, "Optimal Placement of DGs in Distribution System Using an Improved Harris Hawks Optimizer Based on Single and Multi-Objective Approaches", IEEE Access, Vol. 8, 2020, pp. 52815-52829.

[16] M. I. Akbar, S. A. A. Kazmi, O. Alrumayh, Z. A. Khan, A. Altamimi, M. M. Malik, "A Novel Hybrid Optimization-based Algorithm for the Single and Multi-Objective Achievement with Optimal DG Allocations in Distribution Networks", IEEE Access, Vol. 10, 2022, pp. 25669-25687.

[17] M. Purlu, B. E. Turka, "Optimal Allocation of Renewable Distributed Generations using Heuristic Methods to Minimize Annual Energy Losses and Voltage Deviation Index", IEEE Power Energy Society Sector, Vol. 10, 2022, pp. 21455-21474.

[18] M. M. Ansari, C. Guo, M. S. Shaikh, N. Chopra, I. Haq, L. Shen. "Planning for Distribution System with Grey Wolf Optimization Method", Journal of Electrical Engineering Technology, Vol. 15, 2020, pp. 1485-1499.

[19] A. Mohamed, S. Ali, S. Alkhalaf, T. Senjyu, A. M. Hemeida, "Optimal Allocation of Hybrid Renewable Energy System by Multi-Objective Water Cycle Algorithm. Sustainability", Vol. 11, 2019, p. 6550.

[20] M. Khasanov, S. Kamel, M. Tostado-Véliz, F. Jurado, "Allocation of Photovoltaic and Wind Turbine Based DG units Using Artificial Ecosystem-based Optimization", Proceedings of the IEEE International Conference on Environment and Electrical Engineering, Madrid, Spain, 9-12 June 2020, pp. 1-5.

[21] J. S. Chou, D. N. Truong, "A Novel Metaheuristic Optimizer Inspired by Behavior of Jellyfish in Ocean", Applied Mathematics and Computation, Vol. 389, 2021, p. 125535.

Real-Time Fault Identification of Photovoltaic Systems Based on Remote Monitoring with IoT

Original Scientific Paper

Fernando Jacome
Instituto Tecnológico Superior Rumiñahui,
Electricity Career
Sangolquí, Ecuador
fernando.jacome@ister.edu.ec

Henry Osorio
Instituto Tecnológico Superior Rumiñahui,
Electricity Career
Sangolquí, Ecuador
henry.osorio@ister.edu.ec

Luis Daniel Andagoya-Alba
Instituto Tecnológico Superior Rumiñahui,
Electricity Career
Sangolquí, Ecuador
luis.andagoya@ister.edu.ec

Edison Paredes
Instituto Tecnológico Superior Rumiñahui,
Electricity Career
Sangolquí, Ecuador
edison.paredes@ister.edu.ec

Abstract *– The increase in energy demand, as well as the need to protect the environment, has led to the promotion of new forms of generation, including photovoltaic energy. In this scenario, new challenges arise in the field of real-time monitoring of the characteristic variables of this type of system to determine correct operation. This paper presents the methodology of remote monitoring to detect faults in real-time in a photovoltaic system, taking advantage of the variables values that can be obtained from it, and estimating an operating state based on the behavior of these variables. The study used IoT technology for remote data acquisition, and by analyzing them, an estimate of the panel's operating status was made in real time by comparing the values of the variables registered. The study resulted in a real-time remote monitoring system that allows the estimation of the state of operation of a photovoltaic system and the classification of different types of failures that could occur in it. The study concludes that complex monitoring systems can be configured in real-time by technology based on IoT and with an adequate treatment of these variables, it is possible to estimate the photovoltaic systems' state of operation and identify electrical failures in them.*

Keywords*:* *Photovoltaic System, Faults, IoT, Electrical Variables*

Received: July 20, 2023; Received in revised form: December 24, 2023; Accepted: December 29, 2023

1. INTRODUCTION

The increase in demand for energy has caused the photovoltaic energy market to strengthen in recent years, in addition to this, a lot of research has been carried out to increase the efficiency of photovoltaic panels and batteries, thereby reducing the costs of the components of these systems, making the implementation of these generation systems increase their profitable. [1]. With this growth in photovoltaic generation, new challenges are presented for network operators and users who implement these systems, both for self-consumption and for distributed generation. These new challenges are not only related to the installations and the implications that this would present both to the external and internal networks but also to the possible failures that these systems would have in implementation for continuous use, affecting the security and reliability of the network. In this context, new monitoring techniques are necessary to visualize in some way the photovoltaic systems components state. [2].

In [3-5] a classification of all possible faults that could occur in a photovoltaic system is made, among which are those related to line-to-ground contacts, line-to-line contacts, faults due to short circuits, short circuits, etc. internal faults, open circuit faults, arc flash faults, hot spot faults, shading/partial shading faults, bypass diode faults, and degradation faults. Also, four basic principles of fault detection, model-based detection (MBD), real-time detection (RTD), output signal analysis (OSA), and a machine learning technique (MLT) are analyzed. Additionally, mathematical formulations of each failure are detailed, which may be useful for possible analysis of their detection with other types of techniques [3, 4]. In [5] a classification of faults is made according to their location in the photovoltaic system, these may be on the DC side of the system or the AC side. In addition, a failure detection technique is proposed based on the comparison between the results measured in real-time and the prediction results of the model of the efficiency of the photovoltaic array and the inverter to detect energy losses. If the values are lower than the predefined threshold, then the photovoltaic system is considered to be operating normally, otherwise, an anomaly is considered to be present in the system. System efficiency thresholds are established when the system is working under normal conditions without failure. In [6] a review of the multiple failures that could occur in a photovoltaic panel is carried out and a way of classifying them according to the nature of the failure is proposed, having in this classification of failures of physical, electrical or environmental causes. Furthermore, considering the bibliographic reviews carried out in the article, the main characteristics of some fault detection algorithms available for photovoltaic systems that have proven to be effective and feasible to implement are reviewed. The study presents an analysis of some fault detection techniques such as Model-Based Difference Measurement (MBDM), which compares real-time parameters with calculated model data based on the detected instantaneous irradiance and temperature levels to identify system failures. Real-time difference measurement (RDM), which compares real-time values with threshold limits defined based on photovoltaic models or real-time experiments. Output Signal Analysis (OSA) where analysis is applied to the output signal to identify faults, especially transients in the voltage and current waveforms. Machine learning techniques (MLT) where machine learning algorithms are trained to learn the relationship between the input and output parameters of a photovoltaic system and, based on this learning, identify faulty behavior. Infrared thermography (ITH) bases its analysis on the determination of a thermal imbalance in the panel structure; particularly due to the formation of hot spots as a result of some malfunction in the photovoltaic array. In this same way, there is the studio [7] where classification of the faults that can occur in the photovoltaic arrays is carried out and some advanced fault detection techniques analysis is carried out such as Comparison-Based Techniques (CBTs), Statistical and Signal Processing-Based Techniques (SSPBTs), Reflectometry-Based Techniques (RBTs) and Machine Learning-Based Techniques (MLBTs).

The reviews carried out on the monitoring techniques of photovoltaic systems show a constant evolution, from being manual to being carried out through automatic processes using advanced devices and complex processing procedures, both for the acquisition and for the analysis of the data. In [8] a review of the development of some data monitoring techniques for the diagnosis of the state of photovoltaic panels is carried out. The methods are classified into three groups. Manual methods (visual inspection, reflectometry methods, ground capacitance measurements). Semi-automatic methods (thermal cameras, infrared or electroluminescent images for fault location). Automatic methods use data as input to detect failures through algorithms based on modern analysis and prediction techniques such as advanced algorithms and machine and deep learning. Currently, automatic methods have taken a leading role in detecting photovoltaic system faults.

The advantage of these processes is the efficiency for detecting an anomaly in the photovoltaic system as well as the speed of identification and showing them, however, there are still limitations with the fact of going from an experimental context to a real context. This is due to the costs that this type of system implies. In this context, IoT based applications could have a great impact on the development of remote fault detection systems, very useful in places where performing monitoring with other manual techniques is difficult due to the lack of the necessary infrastructure. [9, 10].

The devices used are sensors, Arduinos and Raspberry Pi microcomputers that, depending on the configuration, obtain data from a photovoltaic system which are presented on displays and/or mobile applications or saved in a physical database or a cloud. Initially, these methodologies have been developed for the monitoring of electrical variables, but later they have been used for other applications such as estimates of operating states. [9, 11-14].

IoT-based methodologies have been developed as automatic techniques for fault detection in photovoltaic systems by remotely acquiring data from the panel. In this way, multiple advances have been made in this field through the development of algorithms and prototypes that have allowed the validation of these procedures. In [15] a low cost prototype based on IoT is developed to monitor data from an autonomous photovoltaic system. In this study, current, voltage, temperature, and solar radiation data are monitored.

Through these data, faults related to short circuits, open circuit faults, dust accumulation faults and shading effects are detected. The fault detection process is carried out by comparing the measured magnitudes of voltages and currents with magnitudes calculated through the data obtained from other parameters such as radiation and temperature. The methodology is applied to a laboratory photovoltaic system. The study [16] presents a methodology to detect various types of failures in photovoltaic panels through thermography and artificial intelligence systems. A multilayer neural network is used to identify the type of failure produced in the panel, the input information for neural network training is obtained from data obtained from multiple thermographic analyses. In this same way, studies [17], [18] analyze different methodologies based on Machine Learning (ML) and Ensemble Learning (EL) that could be used to detect complex faults in photovoltaic panels such as multiple faults that most proposed methodologies cannot identify. The study presents a comparative analysis of the possible methodologies to use both ML and EL, for this it uses data from solar panels under certain fault conditions. The studies present the estimates made with these algorithms and their level of precision, concluding that all of them can be used for the proposed purposes. In the studies [19-23] an analysis of the Artificial Neural Networks application is presented depending on the type of failures to be detected, the types of data used, the ANN model and the performance in the diagnosis of failures. Additionally, the study analyzes the challenge of having sufficient data to be able to train the ANNs used. In this way, is recommended to exchange information between researchers and/or research centers to have common data repositories that can serve as input for the realization of this type of predictive model based on ANN.

The main objective of this work is to monitor and estimate a photovoltaic system operation state through real-time measurements, as well as the monitoring of its failures according to the measurements taken directly in the photovoltaic system. For this study, the historical data from the photovoltaic system parameters will be used to determine normal operating intervals of the photovoltaic system and using them to estimate the operating status of the photovoltaic system in different operating scenarios. The process takes as a reference the advances made in the field of fault detection through parameter comparison and the acquisition of remote data in real-time through IoT processes. For this purpose, the process has been divided into three stages: Obtaining and saving data from the photovoltaic system, analysis of data for detection and identification of failures and visualization of results. The main contribution of this project is:

- Integration of a real-time data acquisition process using IoT technology and its use to determine the operational status of a photovoltaic system in real-time through intervals of the system parameters in normal operation.
- The use of historical data generated by the remote monitoring system to determine intervals in the normal operating parameters of the system to determine a fault operating state.
- Identification of faults at specific points of the photovoltaic system based on the comparison of real values in real-time and the intervals of the operating parameters.

With the increase of monitored variables, it has been possible to identify failures not only in the photovoltaic panel but also failures in different points of the photovoltaic system.

2. METHODOLOGY

The methodology is based on a comparative analysis of the variables that describe the photovoltaic system operation. The monitoring of these variables was carried out by a complete remote data acquisition system with IoT technology. The data acquisition procedure is shown in Fig. 1. This document shows the data management by the remote monitoring system to determine the operating status of the photovoltaic system from the comparison of the real-time parameters of the photovoltaic system with the data of the intervals in normal operation, the process is carried out in a Raspberry Pi microcomputer, later these results are sent by the Internet to a display device that can be a mobile or a server.

2.1. OBTAINING AND SAVING DATA FROM THE PHOTOVOLTAIC SYSTEM

The initial stage of the project consists of obtaining the data from the photovoltaic system for its analysis by the developed methodology to estimate the operation state and determine the type of possible failure. This stage consists of the following stage:

Measurement of variables: It is carried out by sensors located at specific points in the photovoltaic system. The variables taken are Voltages, Current, Irradiance and Temperature in the photovoltaic panel and Current in the battery.

Data acquisition and sending: It is done by an Arduino device that is responsible for converting the sensor's analog signals to digital for later sending them by the Raspberry Pi 4 device to a storage cloud through IoT technology.

The stage of obtaining and storing data can be visualized in the graph of Fig. 1.

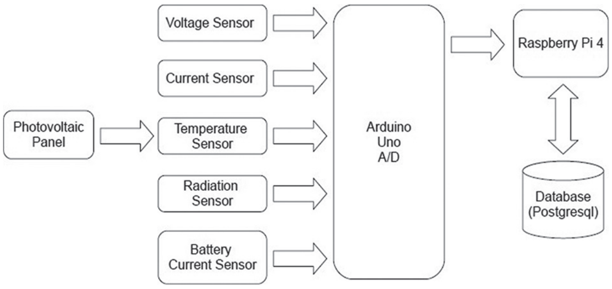


Fig. 1. Monitoring System block diagram using IoT

2.2. DATA ANALYSIS FOR FAULT DETECTION AND IDENTIFICATION

Data processing, for visualization and determining a possible fault in the system, is done by IoT processes directly on the Raspberry Pi, which works as a central node that receives, processes, and sends. the data. Data collected from sensors and monitoring devices are processed in the Raspberry Pi using specific algorithms and models designed to identify patterns of normal and potential failure.

Fault analysis is performed based on measured values. For this purpose, scenarios with different faults and the effect of each of them on the monitored parameters have been analyzed based on their standard values. The monitored variables are Radiation, Solar Panel Voltage, Solar Panel Current, Solar Panel Temperature, and Battery Current. These values represent the optimal system operation in normal conditions, any deviation in the values and according to the analysis proposed in the methodology will represent a possible fault that has occurred in some part of the photovoltaic system.

The data processing from sensors begins through a fixed window width average filtering whose size is ten samples, to eliminate any form of noise that alters the signal coming from the sensors. In addition, the values coming from the PR-300AL solar irradiation sensor were compared using a commercial meter model TES 132. Likewise, the values from the FZ0430 voltage sensor and the ACS712 current sensor were validated using a digital multimeter Proskit MT-3109.

The methodology proposed in the present work focuses on the analysis of seven possible failures in the photovoltaic system.

- Data acquisition process fault.
- Radiation measurement fault.
- Voltage drops in the photovoltaic panel fault.
- Current drop in the photovoltaic panel fault.
- Photovoltaic panel temperature measurement fault.
- General photovoltaic panel fault.
- General battery fault.

The results of this analysis are sent to the results visualization stage.

The data methodology analysis for fault detection can be seen in Fig. 2, in which different faults can be analyzed by simulation of seven possible scenarios.

Data acquisition process fault

In this paper, this scenario has been taken as the first possible fault, if it were to occur, none of the subsequent failures or the actual operating state of the photovoltaic system could be identified. In this fault scenario, it will be considered that none of the variables coming from the monitoring system are within the allowed limits. In this case, a general fault in the photovoltaic system can be considered for some reason that must be verified in the system.

Radiation measurement fault

In the present methodology, radiation measurement failure is considered when the panel voltage, current and temperature variables are within a normal operation, but there is a problem with the radiation measurements.

The main effect of the errors in the measurement of solar radiation occurs in the determination of the performance of the photovoltaic system and in the determination of the climatic and atmospheric conditions that influence the energy production of the photovoltaic system.

Voltage drops in the photovoltaic panel fault

A voltage drop failure scenario in the photovoltaic panel is identified when the radiation and current variables of the photovoltaic panel are within their normal values but the voltage values of the photovoltaic panel present any anomaly.

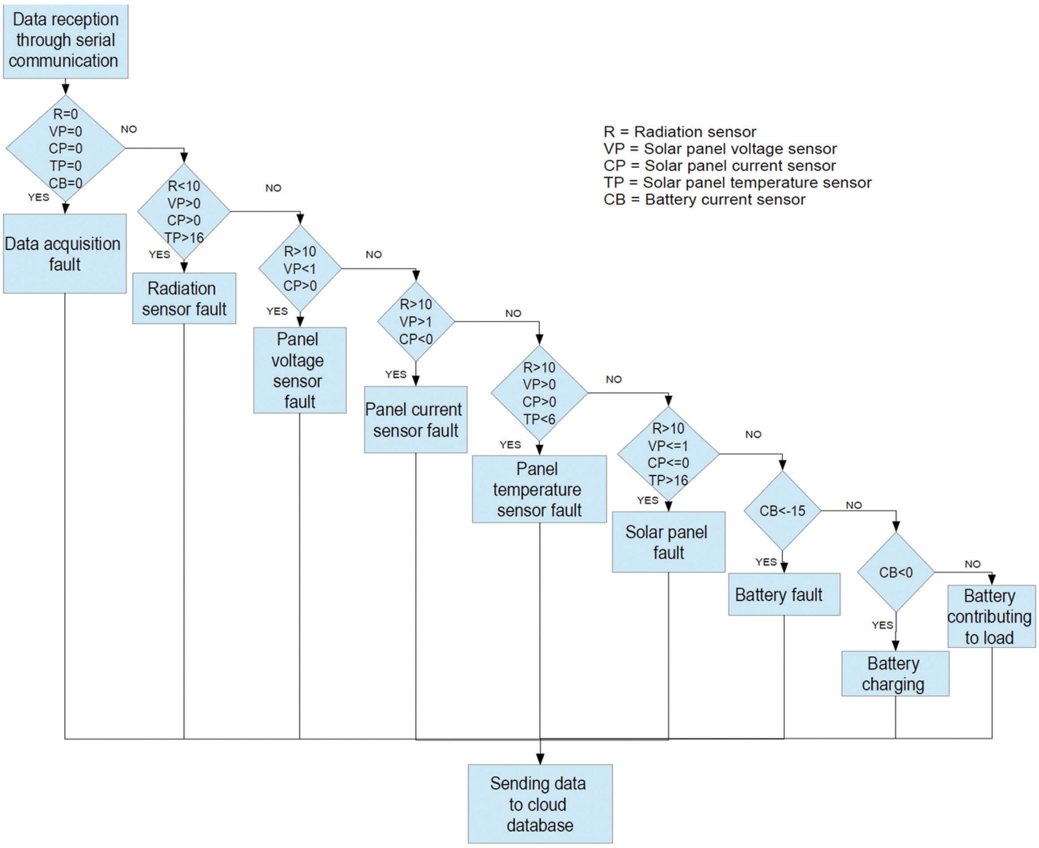

Data reception through serial communication
R=0
VP=0
CP=0
TP=0
CB=0
NO
YES
Data acquisition fault
R<10
VP>0
CP>0
TP>16
NO
YES
Radiation sensor fault
R>10
VP<1
CP>0
NO
YES
Panel voltage sensor fault
R>10
VP>1
CP<0
NO
YES
Panel current sensor fault
R>10
VP>0
CP>0
TP<6
NO
YES
Panel temperature sensor fault
R>10
VP<=1
CP<=0
TP>16
NO
YES
Solar panel fault
CB<-15
NO
YES
Battery fault
CB<0
NO
YES
Battery charging
Battery contributing to load
Sending data to cloud database
R = Radiation sensor
VP = Solar panel voltage sensor
CP = Solar panel current sensor
TP = Solar panel temperature sensor
CB = Battery current sensor

Fig. 2. Diagram of possible faults analysis in a Photovoltaic System

This type of fault is a problem that can occur when the solar panels' generation voltage decreases significantly, which shows that the panels are not producing enough energy even with adequate radiation levels.

Current drop in the photovoltaic panel fault

The fault of the current drop in the photovoltaic panel is a type of fault that can affect the general performance of the photovoltaic system and decrease its energy production. In this methodology, a scenario with a failure by current quality in the photovoltaic panel is identified when the measurements of the current variable are not within their normal operating limits while the radiation and voltage variables of the photovoltaic panel present normal operating values.

Photovoltaic panel temperature measurement fault

The temperature variable is a panel performance indicator. The correct measurement of this parameter can determine the existence of a possible fault in the photovoltaic panel that will affect the energy production of the photovoltaic system. In the present methodology, the identification of some anomaly in the panel temperature is proposed when the measurements of the temperature variables present some anomaly while the values of the radiation parameters, panel voltage and panel current have adequate values.

General photovoltaic panel fault

General faults in photovoltaic panels can be identified in different ways, especially with the decrease in power generation, drops in voltage and output current. To diagnose a general failure, it is necessary to monitor the voltage and current parameters of the photovoltaic panel, as well as the parameters of incident radiation on the panel and its temperature. With these values, a general failure of the panel can be identified and the energy production of the photovoltaic system when all these parameters present some anomaly.

General battery fault

General battery faults can be identified in many ways, such as a decrease in charge capacity or a complete battery fault. To identify this type of fault, a current sensor in the battery has been considered. If the battery current parameter presents null values while other system parameters such as radiation, temperature, panel voltage and current are present.

2.3. RESULTS VISUALIZATION

The treatment of the data for the possible faults visualization in the photovoltaic system is done by processes in IoT, in the case of saving it was done by an open-source relational database AWS-PostgreSQL.

This database management system is known for its open source, reliability, scalability, and ability to manage large volumes of data. AWS-PostgreSQL ensures secure and efficient storage of collected data in the system.

For the visualization of the results, the Qlik Sense application has been used, which is a tool that allows the creation of interactive dashboards and graphs to analyze and visualize the data effectively, with Qlik Sense, this tool allows users to perform analysis, identify patterns, trends, and anomalies, making it easier to spot potential system failures and make informed decisions.

2.4. PHOTOVOLTAIC SYSTEM TEST PROTOTYPE

The methodology has been applied in a real test photovoltaic system located in the Rumiñahui University Institute building in the Pichincha province Fig. 6. The test system consists of the following elements:

- **Photovoltaic panel**: 200W photovoltaic panel, open circuit voltage of 21V and short circuit current of 12.82A
- **Battery unit**: Gel battery, capacity of 100Ah at 12V
- **Charge regulator**: Solar Charge Controller PWM, nominal voltage of 12-24V and maximum current of 20A
- **Voltage Inverter**: 1KW DC/AC inverter, nominal voltage 12VDC/110VAC
- **Current sensors**: ACS712
- **Loads**: 4 9W LED spotlights

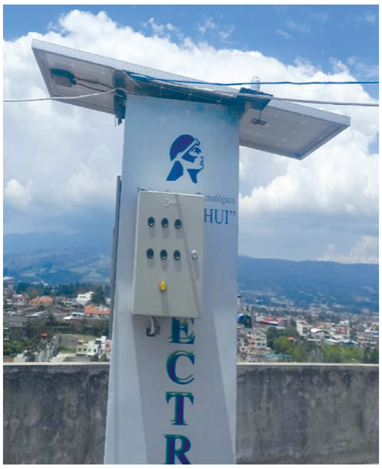


Fig. 3. Photovoltaic system test prototype

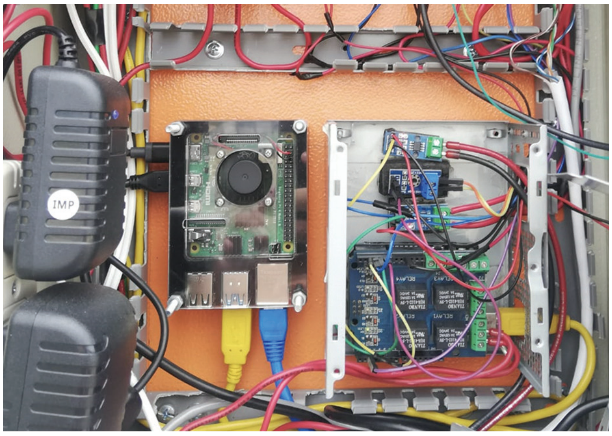


Fig. 4. Raspberry Pi, Arduino, Current sensor and Voltage Sensor

Monitoring and data storage system: This system consists of: Raspberry Pi, Arduino, Current sensor, Voltage Sensor, Temperature Sensor, and Radiation Sensor.

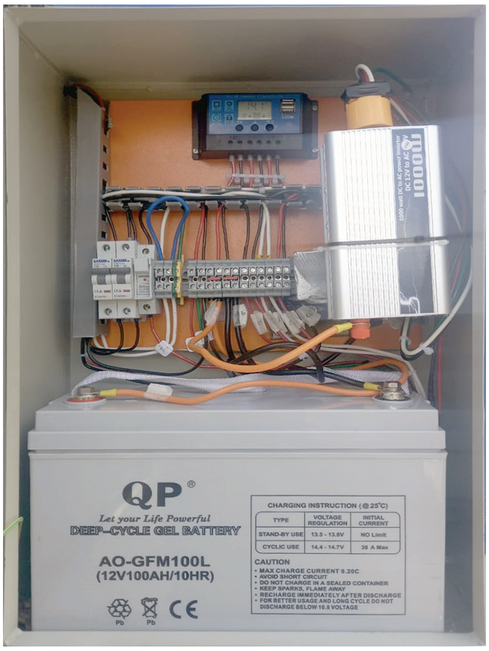


Fig. 5. Photovoltaic system test prototype.

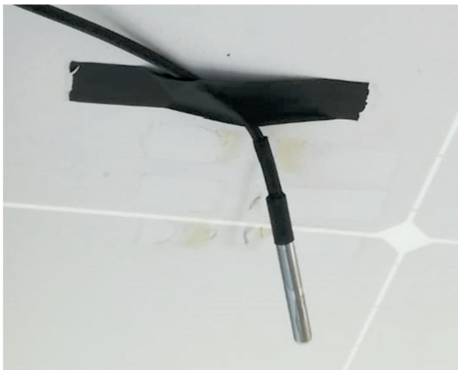


Fig. 6. Temperature Sensor

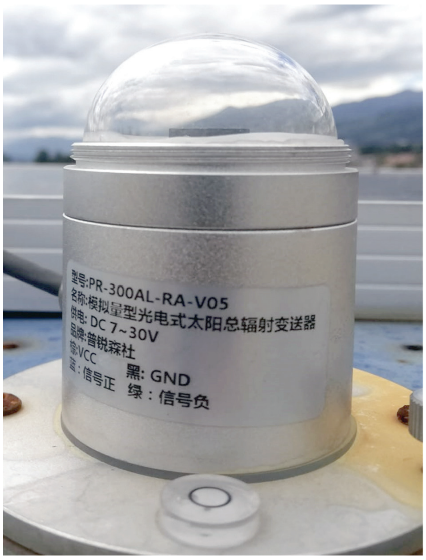


Fig. 7. Radiation Sensor.

The standard values of the parameters monitored in the test photovoltaic system are presented in Table 1.

Component	Value
Radiation	0 - 1000W/m2
Solar Panel Voltage	0 - 21V
Solar Panel Current	0 - 12.82 A
Solar Panel Temperature	0 - 60°C
Battery current	0 - 100A

3. RESULTS

The present work is based on the application of a data acquisition system for monitoring the characteristic parameters of a photovoltaic system. The project is based on IoT technology for the acquisition of data for the supervision of the state of operation of a photovoltaic system, as well as the identification of some possible faults that can occur in some parts of the photovoltaic system. The results of remote monitoring of the variables of the photovoltaic system, as well as its operating state, detail any possible failure in it.

The analysis of all the variables monitored and saving them in the database allows the estimation of the photovoltaic panel operation state. In the case of any anomaly that exists, the comparison of all these parameters will be able to estimate the possible type of fault that has occurred in some part of the photovoltaic system. The methodology applied to achieve this objective is detailed in section II. Here the results obtained from different fault scenarios in the test system are presented.

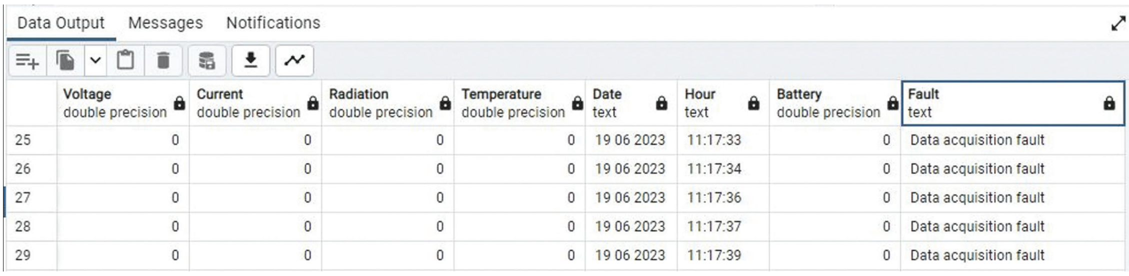
Based on the values indicated in Table 1, and with the application of the methodology proposed in section II in the test photovoltaic system, the following results are obtained.

The Fig. 8. Shows a fault scenario for data acquisition. These types of faults can occur due to technical monitoring devices faults or by human faults due to errors in their connection or configuration.

This scenario can cause many negative effects, such as inadequate monitoring of the photovoltaic system parameters, causing that would not be possible to know its operating state or the existence of a possible fault in any components, which would not present adequate information to users for taking the correct decisions.

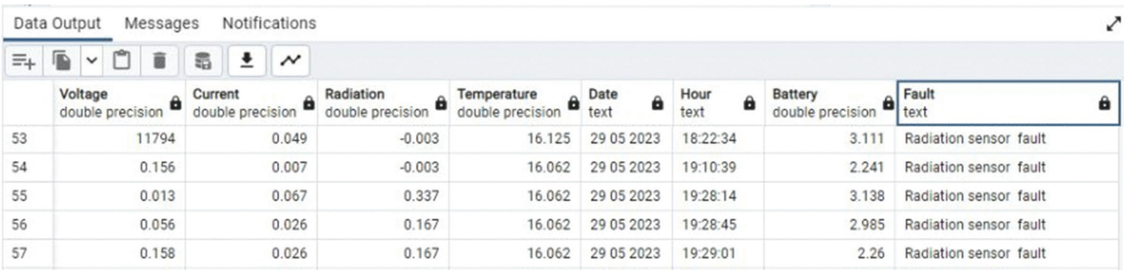
Radiation measurement fault.

The fault in the radiation measurement can be caused by different technical or atmospheric reasons, in the case of technical faults it could be by damage in the measurement sensors due to factory defects or aging, in the case of atmospheric factors it could be by extreme climatic conditions that do not allow the correct radiation measurement by the sensors. The result presented by the graphical interface in a fault event scenario by radiation measurement is shown in Fig. 9.



	Voltage double precision	Current double precision	Radiation double precision	Temperature double precision	Date text	Hour text	Battery double precision	Fault text
25	0	0	0	0	19 06 2023	11:17:33	0	Data acquisition fault
26	0	0	0	0	19 06 2023	11:17:34	0	Data acquisition fault
27	0	0	0	0	19 06 2023	11:17:36	0	Data acquisition fault
28	0	0	0	0	19 06 2023	11:17:37	0	Data acquisition fault
29	0	0	0	0	19 06 2023	11:17:39	0	Data acquisition fault

Fig. 8. Data acquisition process fault messages

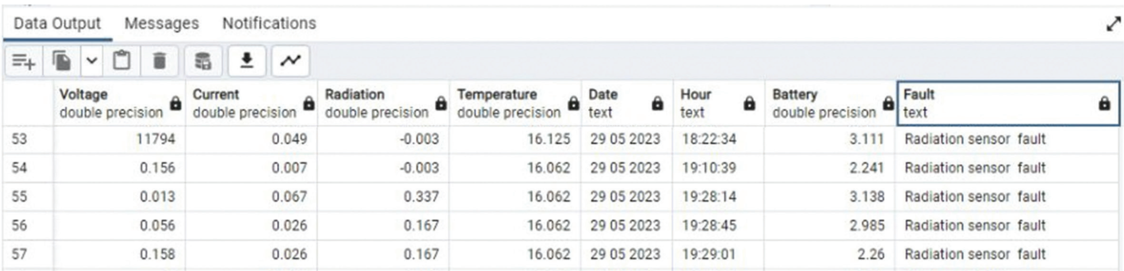


	Voltage double precision	Current double precision	Radiation double precision	Temperature double precision	Date text	Hour text	Battery double precision	Fault text
53	11794	0.049	-0.003	16.125	29 05 2023	18:22:34	3.111	Radiation sensor fault
54	0.156	0.007	-0.003	16.062	29 05 2023	19:10:39	2.241	Radiation sensor fault
55	0.013	0.067	0.337	16.062	29 05 2023	19:28:14	3.138	Radiation sensor fault
56	0.056	0.026	0.167	16.062	29 05 2023	19:28:45	2.985	Radiation sensor fault
57	0.158	0.026	0.167	16.062	29 05 2023	19:29:01	2.26	Radiation sensor fault

Fig. 9. Radiation measurement fault messages

Voltage drops in the photovoltaic panel fault

Faults by voltage drop in the photovoltaic panel can be caused by different factors such as lack of sunlight, partial or total shading of the panels, dirt or dust accumulated on the surface of the panels, faulty connections, cable problems or damage to panel components. These types of faults affect the amount of electrical energy generated and the performance of the photovoltaic system decreases, causing a decrease in the energy production provided by the system. A scenario with fault by voltage drop in the photovoltaic panel is shown in Fig. 10.



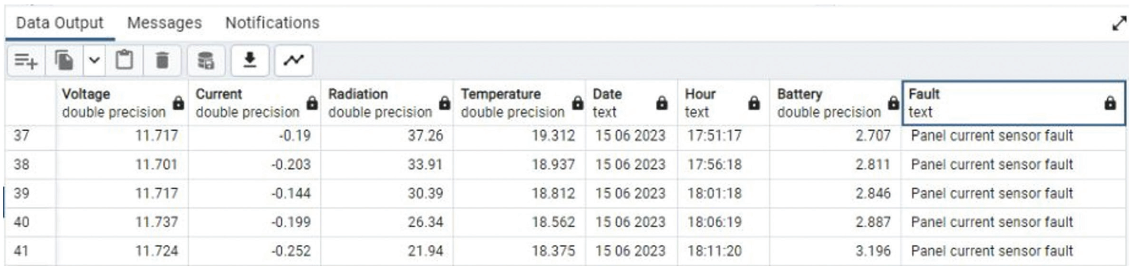
	Voltage double precision	Current double precision	Radiation double precision	Temperature double precision	Date text	Hour text	Battery double precision	Fault text
53	11794	0.049	-0.003	16.125	29 05 2023	18:22:34	3.111	Radiation sensor fault
54	0.156	0.007	-0.003	16.062	29 05 2023	19:10:39	2.241	Radiation sensor fault
55	0.013	0.067	0.337	16.062	29 05 2023	19:28:14	3.138	Radiation sensor fault
56	0.056	0.026	0.167	16.062	29 05 2023	19:28:45	2.985	Radiation sensor fault
57	0.158	0.026	0.167	16.062	29 05 2023	19:29:01	2.26	Radiation sensor fault

Fig. 10. Voltage drops in the photovoltaic panel fault messages

Current drop in the photovoltaic panel fault

A fault by Current Drop in the photovoltaic panel can be caused by different factors, especially the effect of shadows. These faults affect directly the production of the photovoltaic panel and the electricity production of the photovoltaic system. They can also be caused by dirt or dust accumulated on the surface of the panel. Dirt acts as a barrier that blocks sunlight and reduces, the energy output of the panel, resulting in less current being generated. In addition, faulty electrical connections, cable problems, or damage to panel components can also cause a decrease in the generated current.

A scenario with a failure by Current Drop in the photovoltaic panel is shown in Fig. 11.

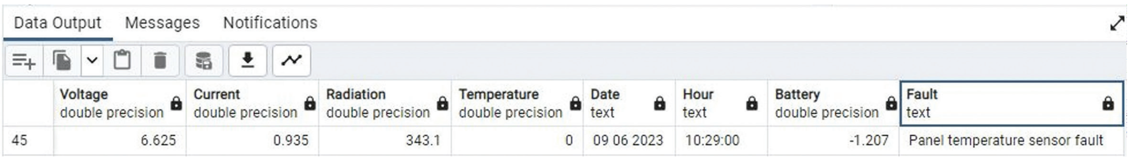


	Voltage double precision	Current double precision	Radiation double precision	Temperature double precision	Date text	Hour text	Battery double precision	Fault text
37	11.717	-0.19	37.26	19.312	15 06 2023	17:51:17	2.707	Panel current sensor fault
38	11.701	-0.203	33.91	18.937	15 06 2023	17:56:18	2.811	Panel current sensor fault
39	11.717	-0.144	30.39	18.812	15 06 2023	18:01:18	2.846	Panel current sensor fault
40	11.737	-0.199	26.34	18.562	15 06 2023	18:06:19	2.887	Panel current sensor fault
41	11.724	-0.252	21.94	18.375	15 06 2023	18:11:20	3.196	Panel current sensor fault

Fig. 11. Current drop in the photovoltaic panel fault messages.

Photovoltaic panel temperature measurement fault

Fig. 12 presents a scenario in which a panel temperature measurement error has occurred. This type of error can be by technical defects in the sensors, due to the existence of some damage in the panel or due to climatic conditions that do not allow them to generate the foreseen energy by the system. In any case, the temperature is a parameter that can show some damage in the photovoltaic system, so it is a variable to be considered.

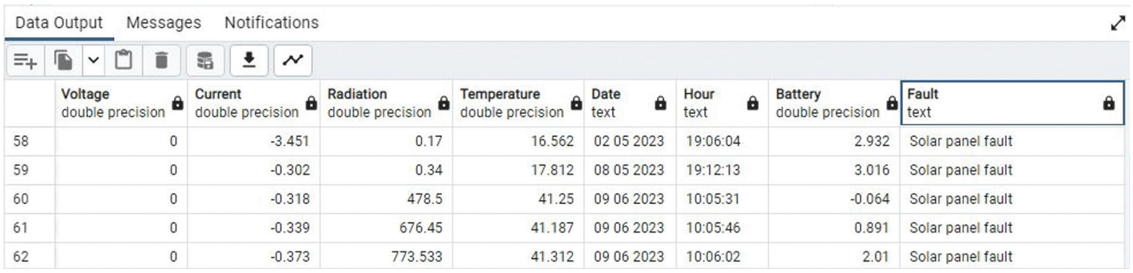


	Voltage double precision	Current double precision	Radiation double precision	Temperature double precision	Date text	Hour text	Battery double precision	Fault text
45	6.625	0.935	343.1	0	09 06 2023	10:29:00	-1.207	Panel temperature sensor fault

Fig. 12. Photovoltaic panel temperature measurement fault messages

General photovoltaic panel fault

A general fault in a photovoltaic panel refers to a general malfunction that affects the energy production of the photovoltaic panel. It can be caused by different factors, such as damage to internal components, aging, damage from adverse environmental conditions, or installation problems. Fig. 13 shows a general failure scenario in the photovoltaic panel.



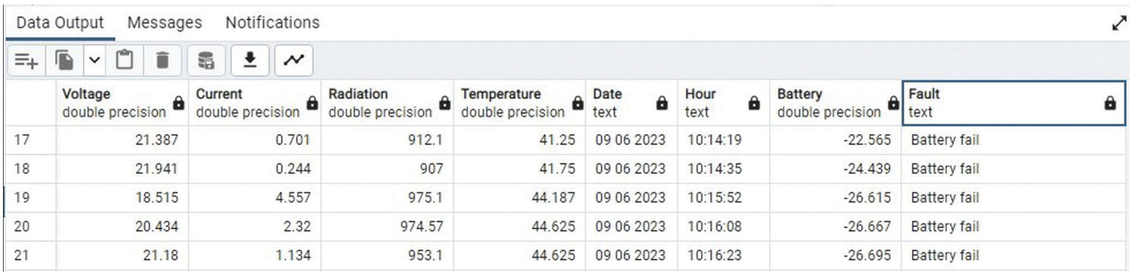
	Voltage double precision	Current double precision	Radiation double precision	Temperature double precision	Date text	Hour text	Battery double precision	Fault text
58	0	-3.451	0.17	16.562	02 05 2023	19:06:04	2.932	Solar panel fault
59	0	-0.302	0.34	17.812	08 05 2023	19:12:13	3.016	Solar panel fault
60	0	-0.318	478.5	41.25	09 06 2023	10:05:31	-0.064	Solar panel fault
61	0	-0.339	676.45	41.187	09 06 2023	10:05:46	0.891	Solar panel fault
62	0	-0.373	773.533	41.312	09 06 2023	10:06:02	2.01	Solar panel fault

Fig. 13. General photovoltaic panel fault messages

General battery fault

A general fault in a battery refers to a general malfunction that affects its storage capacity. It can be caused by various factors, such as battery aging, overcharging, deep discharge, or extreme environmental conditions.

Fig. 14 shows a general fault scenario in the photovoltaic system.



	Voltage double precision	Current double precision	Radiation double precision	Temperature double precision	Date text	Hour text	Battery double precision	Fault text
17	21.387	0.701	912.1	41.25	09 06 2023	10:14:19	-22.565	Battery fail
18	21.941	0.244	907	41.75	09 06 2023	10:14:35	-24.439	Battery fail
19	18.515	4.557	975.1	44.187	09 06 2023	10:15:52	-26.615	Battery fail
20	20.434	2.32	974.57	44.625	09 06 2023	10:16:08	-26.667	Battery fail
21	21.18	1.134	953.1	44.625	09 06 2023	10:16:23	-26.695	Battery fail

Fig. 14. General battery fault messages.

4. CONCLUSIONS

The implementation of systems based on IoT technology used for real-time monitoring of photovoltaic systems has proven to be an effective and promising solution. The ability to remotely monitor, record, save data and analyze it in real-time enables constant monitoring of system variables, making it easy to detect optimal or failed operating states early.

The continuous monitoring of the variables of the photovoltaic system helps to guarantee optimal operation and prevent possible problems caused by failures in the system. The ability to observe the variability of these parameters in real-time provides a complete view of the system and makes it easy to identify significant deviations in operating parameters that could indicate the presence of faults or abnormal conditions. By establishing thresholds and comparison criteria for normal operation, abnormal conditions can be quickly identified and corrective measures can be taken to address the identified failure on time avoiding loss of system efficiency and prolonged damage to photovoltaic system components in general.

The results of this study establish the basis for the development of more advanced and sophisticated real-time monitoring systems in the field of alternative energies. These technologies have the potential to significantly improve the efficiency and reliability of photovoltaic systems, promoting their large-scale adoption and contributing to the transition to cleaner and more sustainable energy sources. In addition, the proposed methodology can be generalized for other types of unconventional generation systems such as wind power, for which it will be necessary to make adaptations to the methodology according to the type of generation to be analyzed.

The results showed that the proposed methodology allows it to be proposed for a large-scale system, however, some points must be considered to achieve this scaling, mainly due to the large amount of data that would be managed, firstly the system should use a scalable IoT platform, which provides the ability to process and collect large amounts of data efficiently. Additionally, a cloud storage system should be implemented, which provides reliable and scalable storage capacity for the data generated by the system, finally, it should be implemented appropriate security measures such as authentication and encryption that help protect the system from possible cyber-attacks.

5. ACKNOWLEDGMENT.

The present document is part of the research project entitled Remote Monitoring of electrical variables and Estimation Faults in photovoltaic systems, financed by the Rumiñahui Universitary Technological Institute.

6. REFERENCES:

[1] G. Masson, E. Bosch, I. Kaizuka, A. Jäger-Waldau, J. Donoso, "Snapshot of Global PV Markets 2022", Task 1 Strategic PV Analysis and Outreach PVPS, 2022.

[2] M. Köntges et al. "Review of Failures of Photovoltaic Modules", International Energy Agency, External final report IEA-PVPS, 2014.

[3] I. U. Khalil et al. "Comparative Analysis of Photovoltaic Faults and Performance Evaluation of its Detection Techniques", IEEE Access, Vol. 8, 2020, pp. 26676-26700.

[4] D. S. Pillai, F. Blaabjerg, N. Rajasekar, "A Comparative Evaluation of Advanced Fault Detection Approaches for PV Systems", IEEE Journal of Photovoltaics, Vol. 9, No. 2, 2019, pp. 513-527.

[5] S. R. Madeti, S. N. Singh, "A comprehensive study on different types of faults and detection techniques for solar photovoltaic system", Solar Energy, Vol. 158, 2017, pp. 161-185.

[6] D. S. Pillai, N. Rajasekar, "A comprehensive review on protection challenges and fault diagnosis in PV systems", Renewable and Sustainable Energy Reviews, Vol. 91, 2018, pp. 18-40.

[8] A. Mellit, S. Kalogirou, "Artificial intelligence and internet of things to improve efficacy of diagnosis and remote sensing of solar photovoltaic systems: Challenges, recommendations and future directions", Renewable and Sustainable Energy Reviews, Vol. 143, 2021, p. 110889.

[7] A. Y. Appiah, X. Zhang, B. B. K. Ayawli, F. Kyeremeh, "Review and Performance Evaluation of Photovoltaic Array Fault Detection and Diagnosis Techniques", International Journal of Photoenergy, Vol. 2019, 2019.

[9] W. Priharti, A. F. K. Rosmawati, I. P. D. Wibawa, "IoT based photovoltaic monitoring system application", Journal of Physics: Conference Series, Vol. 1367, 2019.

[10] S. Muthusamy, "Fault Detection and Monitoring of Solar PV Panels using Internet of Things", International Journal of Industrial Engineering, Vol. 2, 2018, pp. 146-149.

[11] M. P. Tellawar N. Chamat, "An IOT based Smart Solar Photovoltaic Remote Monitoring System", International Journal of Engineering Research & Technology, Vol. 8, No. 9, 2019.

[12] M. Gopal, T. C. Prakash, N. V. Ramakrishna, B. P. Yadav, "IoT Based Solar Power Monitoring System", IOP Conference Series: Materials Science and Engineering, Vol. 981, 2020.

[13] F. Rerhrhaye et al. "IoT-Based Data Logger for solar systems applications", Proceedings of the ITM Web of Conferences, Vol. 46, 2022.

[14] I. Vujović, M. Koprivica, Ž. Đurišić, "Monitoring and management solution for distributed PV systems based on cloud and IoT technologies", Proceedings of the 30th Telecommunications Forum, Belgrade, Serbia, 15-16 November 2022, pp. 1-4.

[15] A. Hamied, A. Boubidi, N. Rouibah, W. Chine, A. Mellit, "IoT-Based Smart Photovoltaic Arrays for Remote Sensing and Fault Identification", Pro-

ceedings of the International Conference in Artificial Intelligence in Renewable Energetic Systems, Tipaza, Algeria, 26-28 November.

[16] A. Haque, K. V. S. Bharath, M. A. Khan, I. Khan, Z. A. Jaffery, "Fault diagnosis of Photovoltaic Modules", Energy Science & Engineering, Vol. 7, No. 3, 2019, pp. 622-644.

[17] A. Mellit, S. Kalogirou, "Assessment of machine learning and ensemble methods for fault diagnosis of photovoltaic systems", Renewable Energy, Vol. 184, 2022, pp. 1074-1090.

[18] C. Kapucu, M. Cubukcu, "A supervised ensemble learning method for fault diagnosis in photovoltaic strings", Energy, Vol. 227, 2021, p. 120463.

[19] B. Li, C. Delpha, D. Diallo, A. Migan-Dubois, "Application of Artificial Neural Networks to photovoltaic fault detection and diagnosis: A review", Renewable and Sustainable Energy Reviews, Vol. 138, 2021, p. 110512.

[20] S. Rao, A. Spanias, C. Tepedelenlioglu, "Solar Array Fault Detection using Neural Networks", Proceedings of the IEEE International Conference on Industrial Cyber Physical Systems, Taipei, Taiwan, 6-9 May 2019, pp.196-200.

[21] X. Lu et al. "Fault diagnosis for photovoltaic array based on convolutional neural network and electrical time series graph", Energy Conversion and Management, Vol. 196, 2019, pp. 950-965.

[22] D. Manno, G. Cipriani, G. Ciulla, V. Di Dio, S. Guarino, V. Lo Brano, "Deep learning strategies for automatic fault diagnosis in photovoltaic systems by thermo-graphic images", Energy Conversion and Management, Vol. 241, 2021, p. 114315.

[23] R. H. F. Alves, G. A. de Deus Júnior, E. G. Marra, R. P. Lemos, "Automatic fault classification in photovoltaic modules using Convolutional Neural Networks", Renewable Energy, Vol. 179, 2021, pp. 502-516.

Artificial Bee Colony Algorithm-based Feature Selection and Hybrid ML Framework for Efficient Rice Yield Prediction

Original Scientific Paper

Manasa Chitradurga Manjunath
Department of Computer Science and Engineering,
Presidency University, Bengaluru,
Karnataka, India
manasacm@presidencyuniversity.in

Blessed Prince Pallayan
Department of Computer Science and Engineering,
Presidency University, Bengaluru
Karnataka, India
blessedprince@presidencyuniversity.in

Abstract – *India's economy predominantly depends on monsoon and agricultural output. Agribusiness products contribute to nearly a quarter of its gross domestic product and 58% of its population depends on agriculture for their livelihood. Certain crops, like rice, are vital to its food security being the most widely grown crop and accounting for one-third production of foodgrains in India. Understanding and enhancing its production is critical in ensuring food availability and promoting sustainable agricultural practices. Rice yield prediction has been a most researched area in the agriculture domain. Machine Learning (ML) frameworks have been found to perform well in patches with large, complex datasets as insufficient feature engineering and temporal dependencies plague efficacy. In this paper, we propose a swam-based meta-heuristic artificial bee colony (ABC) algorithm for feature selection from the dataset sourced from the Agricultural Production and Statistical Division of the Department of Agriculture Cooperation and Farmers Welfare, Government of India. The feature engineering is further optimized by a hybrid model comprising a convolutional neural network (CNN) for learning hierarchical representations and identifying relevant attributes from the complex dataset and long short-term memory (LSTM) for temporal aspects. Finally, a random forest (RF) regressor provides the benefits of ensemble learning, which merges multiple decision trees to remove bias, and variance and improve prediction accuracy. From the results, it is observed that the proposed hybrid model outperforms existing state-of-the-art standalone and hybrid models with the highest coefficient of determination (R^2) and lowest mean square error (MSE) of 0.989 and 13613 respectively. The reliable and efficient hybrid model can aid farmers and policymakers in making informed decisions related to rice yield prediction leading to sustainable agricultural practices.*

Keywords: *Artificial bee colony algorithm, convolution neural network, feature selection, long short-term memory, machine learning, random forest, rice yield prediction.*

Received: November 5, 2023; Received in revised form: December 27, 2023; Accepted: January 25, 2024

1. INTRODUCTION

Rice is an important crop for nations worldwide, including India. Fig. 1 illustrates the annual yield of rice in India from the financial year (FY) 1991-2022 [1]. Its production holds significant importance for several reasons as it provides food security since rice is a staple food for most of the Indian population. Ensuring a high rice yield is crucial for the food security of the country's large and growing population. It is of great economic importance as high rice yields lead to increased agricultural income for the farmers, which, in turn, boosts rural livelihoods and helps alleviate poverty. It has a good export potential too as a robust rice yield not only fulfills domestic demand but also provides a surplus for export. India is one of the world's largest rice producers [2]. Rice exports contribute to foreign exchange and enhance its position in global agricultural trade. It provides employment generation as its cultivation employs a rural workforce, including farmers and laborers involved in planting, harvesting, and processing. A healthy rice yield supports these livelihoods.

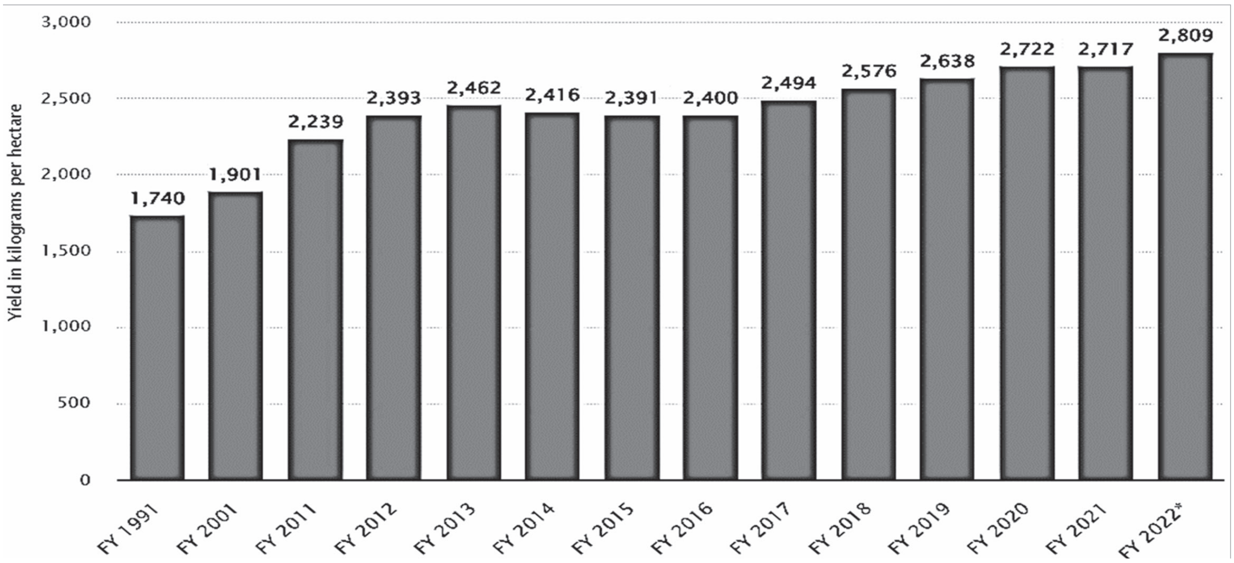


Fig. 1. Annual yield of rice in India FY 1991 to 2022 (in kilograms per hectare) [1]

It is also of big social and cultural significance as rice is deeply embedded in Indian culture, traditions, and cuisine. It is used in many religious rituals and daily meals across the country. It helps in rural development as a strong rice yield encourages investment in agricultural infrastructure, including irrigation systems, storage facilities, and research on improved farming practices. This contributes to rural development and modernization.

Given these reasons, ensuring a sustainable and high rice yield is of utmost importance for India's overall development, food security, and well-being of its citizens. Accurate and timely prediction of its yield can significantly benefit farmers, policymakers, and food distribution systems. By employing advanced computational techniques, such as machine learning, researchers have strived to develop models capable of accurately forecasting rice yields based on various influencing factors. However, this faces several challenges including data availability and quality. High-quality and comprehensive data on several factors that influence rice yield, such as weather conditions, soil characteristics, pest and disease occurrences, crop management practices, and historical yield data, are essential. In many cases, data may be sparse or unevenly distributed across different regions and years, leading to difficulties in building robust and representative predictive models. Rice yield is based on various interconnected factors, including weather patterns, soil health, irrigation practices, and crop management techniques [3]. Capturing the complex interactions and relationships between these variables requires sophisticated modeling techniques. Also, rice is a seasonal crop, and its yield prediction needs to account for seasonal variability, including changes in weather patterns, and pest, and disease outbreaks. The relationships between input variables (e.g., rainfall, temperature, fertilizer application) and rice yield may be non-linear. Thus, traditional linear models may not adequately capture these complex non-linearities. Building complex models to fit the training data too closely can also lead to overfitting, where the model fails to generalize accurately to new, unseen data and does well on training data. Accurate yield prediction often requires historical data over multiple years to identify trends and patterns. In some cases, limited historical data may be available, making it difficult to capture long-term effects accurately. And many advanced machine learning algorithms, such as deep learning models, can be challenging to interpret.

Addressing these challenges requires a combination of domain knowledge, careful feature engineering, model selection, and validation techniques. A proper feature selection technique can have a major influence on the accuracy of rice yield predictions in terms of improved model performance [4-7]. By eliminating noise and irrelevant information, the model can better capture the essential factors that directly influence rice yield, leading to improved prediction accuracy. Feature selection also reduces the complexity of the model, helping mitigate overfitting. A smaller set of relevant features reduces the computational complexity of the model. This results in faster training times and quicker predictions. When the model uses a reduced set of relevant features, it becomes easier to interpret the results. One can gain insights into which specific factors are driving the predictions, and to know the relationships between input variables and rice yield.

Different feature selection methods, such as correlation analysis, recursive feature elimination, feature importance from tree-based models, or advanced techniques like LASSO (L1 regularization), have been used [8-9] but with limited success. The key contributions of this paper are:

- A novel swam-based meta-heuristic artificial bee colony (ABC) algorithm for feature selection from the dataset.

- A reliable and accurate hybrid CNN-LSTM-RF model augmented by feature descriptors from the ABC algorithm for rice yield prediction outperforming existing state-of-the-art standalone and hybrid models.

The manuscript is divided into the following sections: A literature review is covered in section 2. It is followed by Section 3 which outlines the material and methods employed for implementing the proposed model. Section 4 presents the results and compares the performance with existing models. The conclusion is covered in section 5 and references at the end.

2. LITERATURE REVIEW

In the field of rice yield prediction, various techniques have been proposed in the literature. Authors recommend a crop-based prediction system based on site-specific parameters, achieving high accuracy and efficiency [10]. Their recommendation system employs an ML model with a majority voting technique, including RF, Naïve Bayes (NB), Support vector machine (SVM), Linear regression (LR), Decision tree (DT), and XGBoost which motivates to use these ML techniques. They also provide recommendations for suitable fertilizers. The authors in [3] suggested using crop yield projections to optimize fertilizer application. To increase production, they suggested practical fertilizer management practices and employed ML techniques. [2] focuses on predicting paddy yield in the Tamil Nādu Delta region using an MLR-LSTM (Multiple Linear Regression - LSTM) model. This approach aims to provide accurate predictions for paddy yield in this specific region, hence a motivation for using LSTM in a hybrid ML model. [8] employed various ML techniques to analyze and predict crop yields including regression models, DT, SVM, and ensemble methods. Performance evaluation of the best-suited feature subsets for yield prediction is carried out by [11] using ML algorithms. The authors assess different feature combinations and subsets to identify the most influential features for accurate crop yield prediction which motivates us to evaluate computational intelligence (CI) techniques. For capturing complex patterns and data relationships, a hybrid approach using RF and Deep Neural Network (DNN) was proposed by [12]. The results gave better prediction accuracy compared with traditional random forest and deep neural network algorithms, indicating hybrid models are a good fit. A Deep Reinforcement Learning (DRL) model was proposed by [13] for sustainable agricultural applications. DRL combines reinforcement learning techniques with deep learning architectures to optimize decision-making processes and predict crop yields based on environmental factors and other relevant variables. [14] focuses on the most cost-effective means of predicting yields and selecting crops. The study uses artificial neural networks, a reliable tool for modeling and prediction, with forty-six parameters and DNN for crop yield prediction. Both these studies indicate the efficacy of Deep Learning (DL) algorithms compared to ML techniques. IoT is used in [15] to remotely monitor crops with sensor data, and a Multisensor Machine-Learning Approach (MMLA) is proposed for classifying eight crops using the J48 Decision Tree, Hoeffding Tree, and RF algorithms. The RF algorithm proves effective for classifying agricultural text, demonstrating the lowest root mean squared error (RMSE) at 13%, and relative absolute error (RAE) at 38.67%. [16] employ the Normalized Difference Vegetation Index (NDVI) as a crop monitoring tool along with a correlation-based technique to estimate crop production, incorporating physical parameters like soil types and geographic data. They compare SVM, LR, and RF algorithms to improve accuracy and reduce error rates, with RF providing better results. For predicting losses caused by the insect grass grub, [17] uses NB, SVM, DT, RF, NN, K-nearest neighbor (KNN), and ensemble methods. Results from RF and Neural Networks (NN) outperform other classifiers, with ensemble models enhancing the results of weak classifiers. A hybrid model using evolutionary algorithms and data mining techniques is also proposed to enhance findings. All these studies indicate the superiority of RF and evolutionary algorithm efficacy for crop prediction tasks. [18] forecast various crops cultivated in India using the Kernel regression technique, Lasso, and Efficient neural network (ENet) algorithms for yield forecasting, and employ a stacking regression approach to improve algorithms and enhance forecast accuracy, motivating us to employ hybrid approaches.

The ensemble model is employed by [19] aiming to improve the prediction of traits that help overcome hunger-related issues. The study uses a Wild blueberry dataset, utilizing stacking regression (SR) and cascading regression (CR) with a novel combination of ML algorithms. The SR model, with an R^2 of 0.984 and RMSE of 179.898 performed the best. [20] use various data mining techniques to forecast crop production and summarize different crop prediction approaches with different machine learning algorithms. [21] highlight the relevance of remote sensing-based techniques for estimating crop output, comparing remotely derived datasets to in-field survey-based data. [22] introduce the eXtensible Crop Yield Prediction Framework (XCYPF) to predict agricultural yields in precision agriculture, combining relevant indices with information on rainfall and surface temperature for rice and sugarcane crop yield prediction. These studies use datasets with many complex parameters thus inducing the need for better feature selection techniques. [23] propose DL methods like CNN and LSTMs for strawberry yield prediction five weeks ahead, also utilizing yield and weather input data to predict strawberry prices. Their Attention (ATT) based CNN-LSTM model outperforms other ML and DL models for both yield and price prediction using weather data. Thus, CNN-LSTM is a good combination if used in a hybrid mode. [24] use an ensemble model with a majority voting technique including NB, RF, Chi-square Automatic Interaction Detector (CHAID), and KNN as learners for a crop recommendation system that accurately and efficiently suggests crops for site-specific parameters. [3] aim to optimize fertilizer application based on crop yield predictions, using ML techniques, and proposing effective fertilizer management strategies to enhance productivity. [25] develops an accurate prediction model for rice yields by utilizing ML algorithms, specifically focusing on rice crop prediction, and presenting a framework that integrates various ML models for improved accuracy. [26] integrates ML techniques with Streamlit, a user-friendly interface, allowing users to analyze and predict crop production effectively, emphasizing the practical implementation of ML models. [27] proposes a comprehensive approach for predicting crop yield using hybrid ML algorithms, combining various techniques to efficiently predict rice crop yield by integrating factors such as weather information, soil properties, and historical yield data. [28] investigate the use of data mining (DM) techniques for crop yield prediction, employing ML algorithms to explore historical crop data and predict future yields, highlighting the importance of data mining in improving prediction accuracy. All these studies again provide the effectiveness of ML and hybrid techniques for crop yield prediction. [29] focus on rice crop yield prediction using Artificial neural networks (ANN), developing an ANN-based model to forecast rice yields based on various input parameters, highlighting the effectiveness of DL techniques in predicting rice crop yields and their potential for enhancing agricultural decision-making. [30] conduct a study on crop analysis and seed marketing, using regression and association rule mining techniques to analyze crop data and identify patterns. Filter, Wrapper, and embedded methods are some of the feature selection methods used by [5, 9], helping farmers make more informed decisions about crop management and improving yields. Authors in [31] used Feature shuffling and Feature performance feature selection methodology with a hybrid model comprising DT, XGBoost, and RF achieving a coefficient of determination (R^2) of 98.6. Filter methods evaluate features independently of the classification model and rank them based on their correlation or mutual information with the target variable [6]. Wrapper methods evaluate feature subsets by repeatedly training and evaluating a classification model on different subsets. They search for the optimal subset of features that gives the best model performance but can be computationally expensive [31]. Embedded methods integrate feature selection into the model training process itself [32]. These studies indicate the need for optimal feature selection methodologies to reduce computations but at the same time not affecting accuracy. However, the use of computational intelligence (CI) or genetic algorithms for feature selection in crop yield prediction has been limited in the literature.

3. MATERIAL AND METHODS

The overall block diagram of the proposed model is shown in Fig. 2. Feature selection plays a vital role in rice yield prediction as it enhances model performance, reduces dimensionality, improves interpretability, and optimizes resource allocation. By identifying the most relevant features, predictive models can provide accurate and actionable insights to aid farmers, agricultural researchers, and policymakers in making informed decisions and achieving higher rice yield sustainably. We propose a CI algorithm-based feature selection technique. It uses the Artificial Bee colony (ABC) algorithm for selecting optimum features.

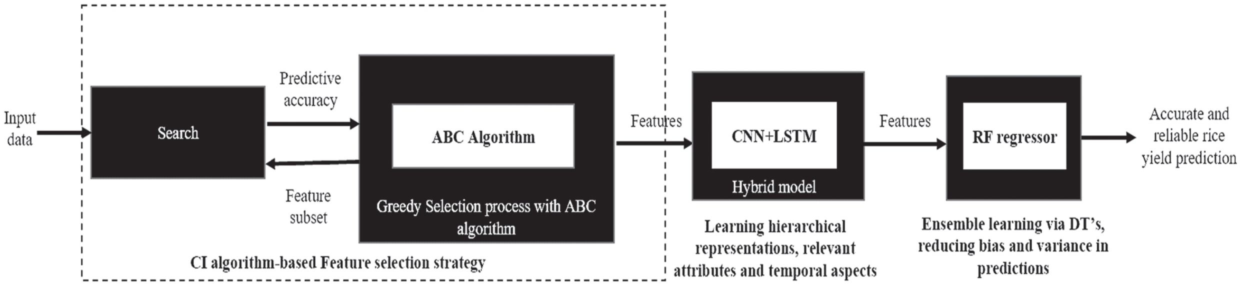

Input data
Search
Predictive accuracy
Feature subset
ABC Algorithm
Greedy Selection process with ABC algorithm
CI algorithm-based Feature selection strategy
Features
CNN+LSTM
Hybrid model
Learning hierarchical representations, relevant attributes and temporal aspects
Features
RF regressor
Ensemble learning via DT's, reducing bias and variance in predictions
Accurate and reliable rice yield prediction

Fig. 2. Block diagram of the proposed framework

3.1. DATASET

The dataset used in this study was sourced from the Agricultural Production and Statistical Division of the Department of Agriculture Cooperation and Farmers Welfare, Government of India https://data.gov.in/sector/Agriculture. The dataset consists of seven features such as state, district, production, year, season, area, and the target variable, yield. Exploratory data analysis (EDA) is conducted to study the dataset characteristics.

3.2. ARTIFICIAL BEE COLONY (ABC) ALGORITHM

The ABC algorithm, introduced by Karaboga [33-35], is a stochastic optimization technique based on the foraging behavior of honeybee swarms. This method is versatile and can be applied to various tasks such as classification, feature selection, clustering, and optimization. The algorithm's flowchart is depicted in Fig. 3. We leverage the ABC algorithm as a tool for feature selection, drawing inspiration from the natural behavior of honeybees in their colonies. Honeybees exhibit impressive communication, coordination, and self-organization skills in their foraging activities. Communication is facilitated through a behavior known as the 'waggle dance,' which effectively directs other bees to fruitful food sources. In this context, a swarm of bees, denoted as 'S' bees (forming the population), is established.

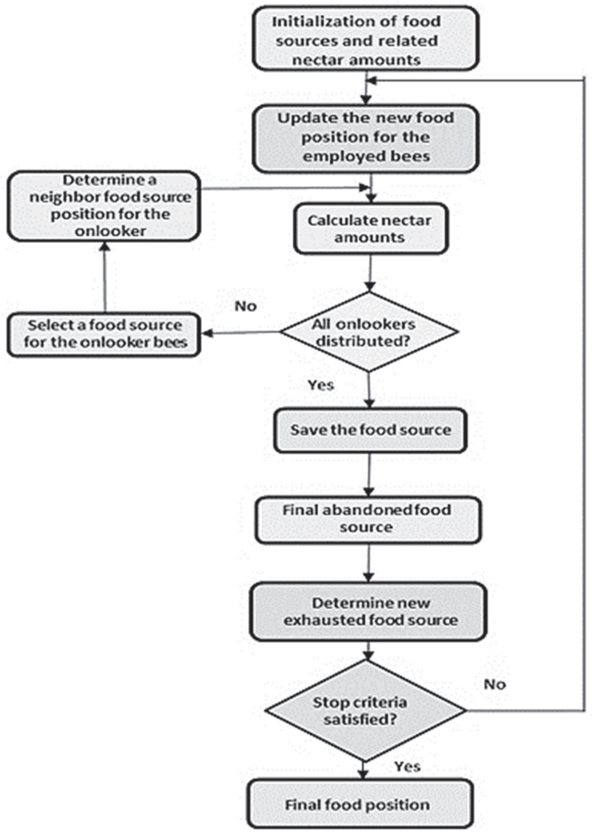

Initialization of food sources and related nectar amounts
Update the new food position for the employed bees
Determine a neighbor food source position for the onlooker
Calculate nectar amounts
No
Select a food source for the onlooker bees
All onlookers distributed?
Yes
Save the food source
Final abandoned food source
Determine new exhausted food source
No
Stop criteria satisfied?
Yes
Final food position

Fig. 3. Flow-chart of ABC algorithm

Potential solutions are represented as food sources, assigned to the bees in a d-dimensional space, aligning with the number of parameters in the optimization problem. The fitness (f_i) metric measures the quantity of nectar at a given food source. The honeybee colony consists of three groups: the employed bees (EB), the onlooker bees (OB), and the scout bees (SB). In each cycle, the algorithm proceeds in the following steps for all the categories of bees.

- Onlooker bees (OB) refer to bees waiting on the 'dance floor' inside the hive.
- In the first cycle, EBs move to random food sources, evaluate nectar amounts, and share this information with OBs.
- OB uses a greedy selection process based on the nectar information received from EB to update their positions.
- In the next cycle, EB selects new food sources in the vicinity of the sources found in the previous cycle, using their memory, and compares nectar amounts.
- OBs use this information to select food sources based on nectar value, and the probability of selecting a position increases with higher nectar amounts.

EB and OB both engage in the search for improved food positions during each cycle. If the nectar quantity at a food source does not improve within a limited number of cycles, the bees abandon those positions. New food sources are randomly generated and assigned to bees called SBs, effectively creating new sources of food. Initially, OBs and SBs are considered unemployed bees (UBs). The exploitation of nectar in food sources is primarily conducted by EB and OB. In every cycle, the food source with the highest present nectar quality is memorized. This food source's position represents the local optimal solution for that cycle. The entire process of food source search, involving EB, OB, and SB, is executed for a specified number of cycles (*k_max*). The best global solution among all the cycles yields the optimum solution.

The step-by-step procedure is as follows:

- Initialization step: Algorithm control parameters are set, including population, dimension, maximum cycles (*k_max*), and limits (*x_min* and *x_max*). Another limit (B) is established to determine when a food source should be abandoned if further improvement or exploitation is not possible.
- Employed Bee (EB) search: EB searches its neighborhoods, utilizing a greedy selection process to choose between the current food source and one within the neighborhood. If the new source is superior, it updates its position (x_{iD}) accordingly; otherwise, it retains the current position.
- Onlooker Step: Each OB is assigned a probability (*Pi*) proportional to the quality of the selected food source, as determined by Eq. (1).

$$P_i = \frac{f_i}{\sum_{i=1}^{s} f_n} \quad (1)$$

A new candidate position is generated based on existing memory, as represented in Eq. (2).

$$v_{ik} = x_{ik} + \in_{ik} (x_{ik} - x_{oi}) \quad (2)$$

values o= {1, 2, 3…N} and k= {1, 2, 3…D} are randomly assigned $\in_{ik}$ is a random number chosen from the range [-1, 1]. If the new solution's value exceeds *x_min* and *x_max*, it is adjusted to the acceptable limits, and its fitness is evaluated.

- Scout Step: To abandon a food source, a control parameter (B) is utilized. If a predetermined number of trials (T) surpasses B, the food source is abandoned, and SBs are generated. New food source positions are discovered by SBs, and existing food positions are updated randomly using Eq. (3):

$$x_i^j = x_{min}^j + rand(0,1)(x_{max}^j - x_{min}^j) \quad (3)$$

The ABC algorithm acts as a cluster to choose optimal features. At the end of the training, feature vectors named *food* about each class are obtained. These obtained features are fed into the hybrid CNN+LSTM model.

Table 1. ABC algorithm initialization parameters

Parameter	Number	Remarks
Population (S)	30	Max. population of bees
Food no.	15	No. of sources of food. ~50% of population
Limit (B)	100	Max. limit after which source of food is abandoned and cannot be further improved
No. of iterations	100	No. of foraging cycles
Runtime	25	No. of runs to see its robustness

3.3. CONSTITUENTS OF THE HYBRID MODEL

The CNN model is used to capture relevant informative patterns and relationships between regions and their impact on rice yield. CNN is effective in automatically learning hierarchical representations and identifying relevant attributes from complex datasets. This method is extremely valuable in dealing with diverse input features, which may have non-linear relationships with rice yield. By applying non-linear transformations and feature extraction, the CNN component captures intricate relationships and patterns that may not be evident through traditional linear models.

Next, the LSTM is tuned to manage the temporal aspect by capturing long-term dependencies in sequential data. The LSTM model is well-suited to manage variable-length sequences and effectively structure the temporal dynamics of rice growth. It can manage the input data with varying time steps and learn the patterns and dependencies present in the sequential data.

Finally, the RF regressor provides the benefits of ensemble learning, which merges multiple decision trees to improve prediction accuracy. Incorporating the RF regressor into the hybrid model provides the advantage of its ability to manage non-linear relationships, manage missing data, and provide robust predictions. Additionally, the RF provides interpretability by offering feature importance rankings, enabling us to identify the most influential variables contributing to the predictions. The RF ensemble further reduces bias and variance, leading to more reliable and accurate predictions for rice yield. The RF regressor complements the CNN-LSTM architecture by adding diversity and reducing prediction errors, leading to more reliable predictions for rice yield.

The idea behind the proposed hybrid model is to offer flexibility in incorporating several types of data sources, allowing us to include various features relevant to rice yield prediction. This flexibility enables the model to adapt to different datasets and capture domain-specific knowledge.

3.4. MODEL FLOW

The complete model flow is explained in Fig. 2. We initially use the ABC algorithm to capture optimal features suitable for the prediction and identifying the target variable. The selected features data, though optimal, are pre-processed to manage any anomalies like missing values and inconsistency. The processed data is fed as input to the hybrid CNN-LSTM model as shown in Fig. 2. The CNN model consists of three Conv1D layers which take the input of the pre-processed data comprising 512, 256, and 128 filters respectively. These layers are followed by two LSTM layers consisting of 100 units each. Three dense layers incorporate 512, 10, and 1 output units, respectively. These dense layers are responsible for structuring the output for making accurate predictions. The output from the combined model is fed to the RF regressor which as mentioned in section 3.3 offers the benefits of ensemble learning for reliable predictions. And finally, using the standard metrics the performance of the entire model is evaluated.

3.5. CONSTRUCTING THE MODEL

3.5.1. Import necessary package

The NumPy library in Python is used for managing the numeric data. Pandas package is used for data manipulation and restructuring. Matplotlib is used to visualize the data in the form of graphs. The scikit-Learn library consists of many key features and plays a critical role in pre-processing and getting the data ready to be fed into the model. TensorFlow Keras library manages the deep learning models.

3.5.2. Feature Selection and Pre-Processing

Feature selection is the most important part of constructing a model. For this, we have used the ABC algorithm as explained in section 3.2 and we capture the model performance with and without ABC feature selection. In the pre-processing stage, we employ label encoding to manage string values in four dataset columns, assigning unique numerical labels to each category. This enables effective processing and analysis of categorical data. Further, we utilize the min-max scaler to ensure feature equalization and compatibility. Scaling the numerical features within a specific range eliminates bias and discrepancies caused by measurement unit differences, preserving relative relationships between values. Incorporating label encoding and min-max scaling techniques transforms the dataset, enabling our hybrid model to manage categorical variables and achieve balanced feature representation.

3.5.3. Implementing and training the model

The CNN-LSTM model is implemented using the Sequential API. The model as presented in Fig. 4 consists of three Conv1D layers for feature extraction in CNN, two LSTM layers for temporal sequence modeling, and three dense layers of output units for shaping and presenting the output. This nature of the hybrid model makes it completely capable of identifying and extracting the hidden features present in the dataset. The model is trained on the dataset using the *fit*() function. The inclusion of the early-stopping call-back method is for stopping the epochs to prevent overfitting. This trained model is used to make predictions on the train and the test data, and the extracted characteristics features feed the RF regressor. The model parameters of the CNN-LSTM model are displayed in Fig. 5. In the summary the first column contains the names of the layers present in the architecture. The output shape indicates the output tensors generated by the model, it comprises (*batch_size*, *timesteps*, and *filters/units*). The *Param #* column contains the number of trainable parameters that are present in each layer. The RF model is built with modified hyperparameters as shown in Table 2.

The *n_estimator* is used to define the number of trees that will constitute the RF model, *max_depth* controls the maximum depth of each decision tree, the *min_sample_split* is the minimum number of nodes that will be present before splitting the nodes and *min_sample_leaf* tells us about the minimum number of leaf's that will be present in each node. Finally, the Random Forest regressor is trained using the extracted attributes from the CNN-LSTM model to obtain the final output.

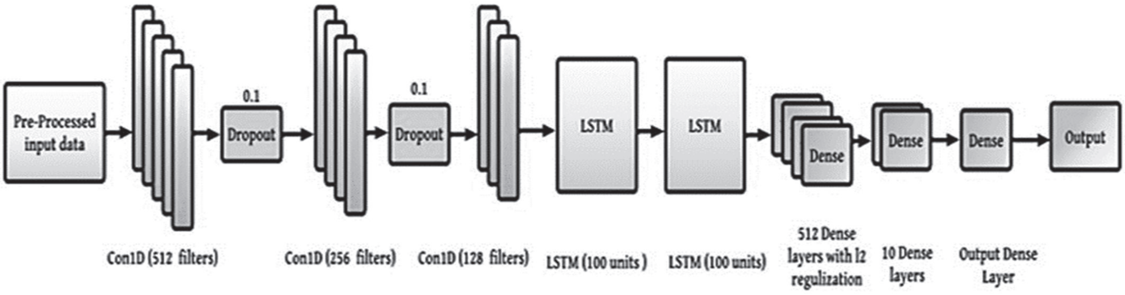


Fig. 4. Block diagram of CNN-LSTM model

3.5.4. Evaluation parameters

The evaluation parameters used in this study are Mean Square error (MSE), Mean absolute error (MAE), Mean absolute percentage error (MAPE), coefficient of determination (R^2), and Root mean square error, calculated using Eq. (4), (5), (6), (7), and (8) respectively.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}) \quad (4)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}| \quad (5)$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|A_i - F_i|}{A_i} \quad (6)$$

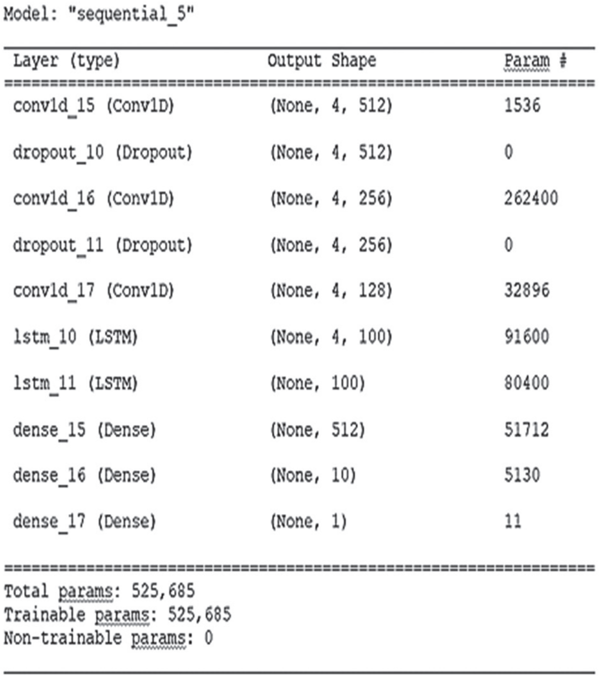
$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2\ (sum\ squared\ regression)}{\sum(y_i - \bar{y})^2 (total\ sum\ of\ the\ squares)} \quad (7)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})} \quad (8)$$

where n=total number of values, y_i=actual value, ŷ=predicted value, A_i=actual value and F_i=predicted value.

Table 2. Random Forest parameters

Parameter	Value
Number of estimators	2000
Max. Depth	6
Min. samples split	5
Min. samples leaf	3



```
Model: "sequential_5"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_15 (Conv1D)           (None, 4, 512)            1536

dropout_10 (Dropout)         (None, 4, 512)            0

conv1d_16 (Conv1D)           (None, 4, 256)            262400

dropout_11 (Dropout)         (None, 4, 256)            0

conv1d_17 (Conv1D)           (None, 4, 128)            32896

lstm_10 (LSTM)               (None, 4, 100)            91600

lstm_11 (LSTM)               (None, 100)               80400

dense_15 (Dense)             (None, 512)               51712

dense_16 (Dense)             (None, 10)                5130

dense_17 (Dense)             (None, 1)                 11

=================================================================
Total params: 525,685
Trainable params: 525,685
Non-trainable params: 0
_________________________________________________________________
```

Fig. 5. CNN-LSTM model parameters

4. RESULTS AND DISCUSSION

The results of EDA performed on the dataset are first discussed. Table 3 illustrates a cross-section of the dataset used as mentioned in section 3.1.

Next, we examine the basic statistical properties of the data. We calculated the count of the data, unique values in columns, and descriptive statistics, such as mean, standard deviation, minimum, maximum, and quartile values for each numerical feature, namely, Area (A), Production (P), and Yield (Y). These statistics helped us understand the range and distribution of values within each variable. The statistical data is represented in Table 4 and Table 5.

Table 3. Cross section of the crop (rice) dataset

State	Dist.	Year	Season	Area	Prod	Yield
Bihar	Purnia	2001-02	Autumn	77801	63333	0.814
Bihar	Purnia	2001-02	Kharif	20748	43473	2.095
Bihar	Purnia	2001-02	Summer	21846	38924	1.781
Bihar	Purnia	2002-03	Autumn	20792	24332	1.170

Table 4. Statistics for categorical data

	State	District	Year	Season
Count	21611	21611	21611	21611
Unique	36	694	25	6
Top	Uttar Pradesh	Purulia	2019-20	Kharif
Frequency	2142	69	1153	9831

Table 5. Statistics of numerical values

	Area	Production	Yield
Count	21611	21611	21611
Mean	46079.02	103475.4	2.09
Std	65787.41	171058.4	1.01
Min	0.35	0	0
25%	2787	4211.5	1.34
50%	16810	29000	2.04
75%	66093.5	129257.5	2.7
Max	687000	1710000	22.37

Next, we explore the categorical variables, namely, State (St), District (D), Year (Y), and Season (S). We examined the unique values present in each category and assessed the frequency distribution of these values. The values are shown in Table 6.

Table 6. Statistics of categorical variables

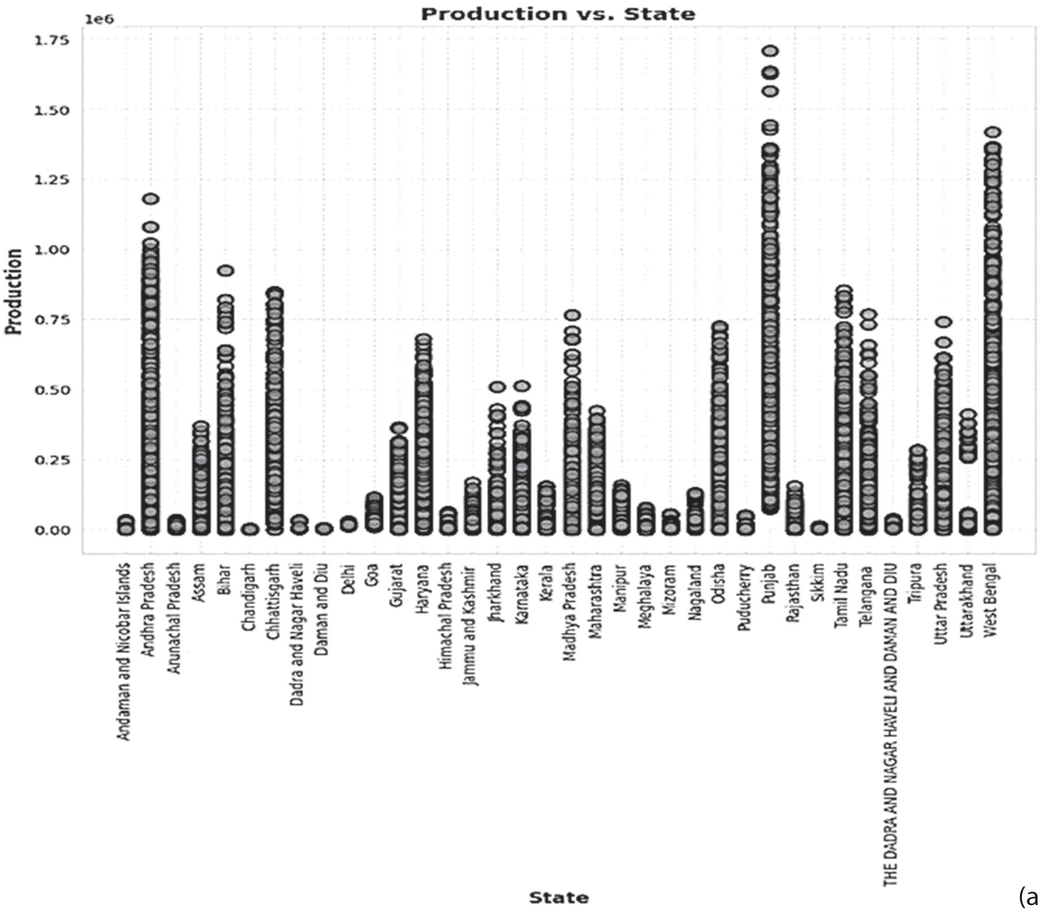
Feature	Unique values
State	36
District	694
Season	6
Year	25

To gain further insights, we visualized the data using various graphical techniques. Fig. 6 shows the distribution of rice cultivation across different states.

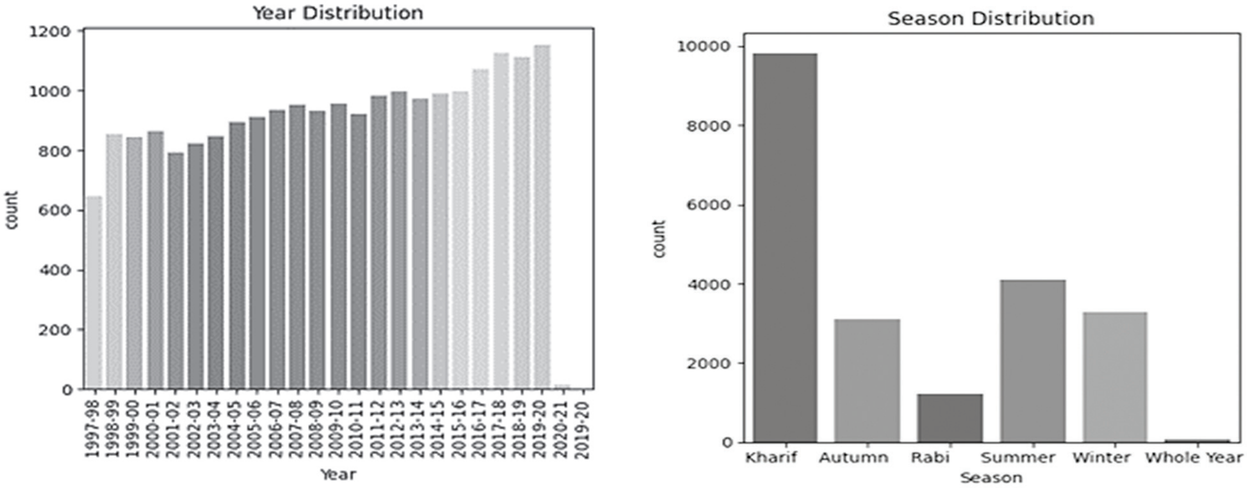
The result of the dataset description post using the ABC feature selection (FS) algorithm is shown in Table 7. It chooses five out of the initial seven features optimally.

Table 7. Dataset description after ABC feature selection

FS method	St	D	Y	S	C	A	P
ABC algorithm			*	*	*	*	*



(a)



(b)

(c)

Fig. 6. Results of exploratory data analysis; (a) State vs. Production for rice yields; (b) Year vs. Production for rice yields and (c) season vs. production for rice yields.

Also, the proposed model outperforms existing state-of-the-art models with an RMSE of 116.67, MAE of 7.43, RAE of 8.67, MAE of 7.43, MSE of 13613 and R^2 of 0.989. MAE scores provide information about the difference between actual and predicted values. A lower MAE value indicates a more efficient model in predicting yield. The MSE score indicates the squared difference between the true and predicted numbers. Computing the RMSE score helps measure the standard deviation of the residuals. The lower the MSE and RMSE scores, the better the model for calculating returns. The higher R^2 value of the proposed model indicates that it captures the spatiotemporal non-linear features well and uses them effectively in predicting the yield.

A comparison between the actual and anticipated values in the case of the proposed hybrid model is shown in Fig. 7. This graph provides a visual representation of the model's performance capturing the patterns and trends of the rice yield data. Since our model has a high R^2 value, the close alignment between the true & predicted values demonstrates a good model fit.

Table 8 presents the scaling characteristics of the LSTM model. Scaling characteristics are measured by increasing the number of epochs and *max_depth* by parameter hyper-tuning. From the results, it is observed that as both these values are increased, the R^2 value too increases, with the best value achieved at 50 epochs, but the time taken also increases. This time can be decreased by using graphical processing units (GPU) as part of future research.

The result of the hybrid model implemented with and without the ABC feature selection technique and comparing its performance with existing state-of-the-art models is captured in Table 9. ABC FS methodology along with the hybrid implementation of CNN+LSTM and RF regressor helps in improving the model performance compared to its performance without any feature selection.

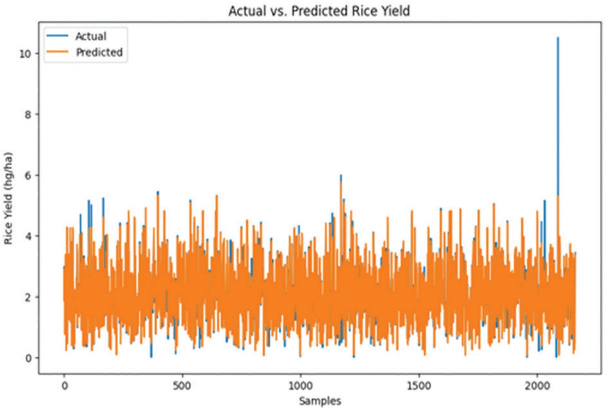


Fig. 7. Actual vs. predicted values of rice yield

Table 8. Scaling characteristics of the LSTM model

Epochs	max_depth	Total_time (mins)	R^2
5	6	15	0.9833
10	6	17	0.9830
20	6	25	0.9824
50	6	40	0.9844
5	8	18	0.9886
10	8	25	0.9878
20	8	42	0.9850
50	8	58	0.9899

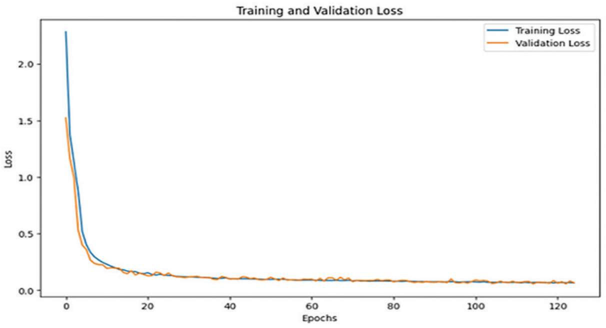


Fig. 8. Training and validation loss in the CNN-LSTM model

Table 9. Comparison of proposed hybrid model with existing models (best results in **bold**)

Reference	Method	R	RMSE	RAE	MAE	R^2	MSE
[25]	K-Star	0.95	365.22	26.65	223.43	0.910	133386
	LR	0.55	936.57	79.57	666.96	0.302	877163
	Gaussian process	0.72	790.54	65.64	548.63	0.525	300995
	MLP	0.76	760.77	68.29	572.48	0.588	327733
	RBF	0.09	1117.03	100.09	839.01	0.008	1247756
	Bagging Model	0.79	700.11	55.46	464.89	0.625	490154
	Additive Regression	0.53	949.21	81.21	680.73	0.285	900999
[26]	Rigid Regression	NA	NA	NA	NA	0.542	NA
	Gradient Boosting	NA	NA	NA	NA	0.667	NA
	Random Forest	NA	NA	NA	NA	0.967	NA
	XGBOOST_ Regression	NA	NA	NA	NA	0.867	NA
[28]	J48	NA	0.27	37.97	0.11	NA	NA
	LWL	NA	0.32	76.34	0.22	NA	NA
	LAD Tree	NA	0.41	68.88	0.19	NA	NA
	IBK	NA	0.30	35.86	0.10	NA	NA
[31]	Hybrid DT, XGBoost, RF with Feature shuffling and Feature performance	0.11	335.10	11.29	8.6023	0.879	112296
[36]	Random Forest	NA	NA	5.82	15	NA	NA
	K-star	NA	NA	12.8	34	NA	NA
	Bays-net	NA	NA	68.45	18.5	NA	NA
	J48	NA	NA	59.29	16	NA	NA
Proposed Model	CNN+LSTM+RF	0.98	122.7	13.1	5.57	0.98	15065
	CNN+LSTM+RF with features selected by the ABC algorithm	**0.99**	116.67	8.67	7.43	**0.989**	13613

The training versus validation loss for the CNN-LSTM model is shown in Fig. 8. which demonstrates the learning advancement of our model during the training process. The graph shows the convergence of the model with reducing training loss, and the low value of validation loss compared to training loss confirms the effectiveness of the training procedure.

Overall, with a high R^2 value of 0.989 and, a low MSE value of 13613 with optimal features selected by the ABC algorithm, and proposed hybrid CNN+LSTM model along with the RF regressor successfully captures the complex relationships between the input features and crop yield. It suggests that this hybrid model can be relied upon to provide accurate predictions of crop yields, which can be invaluable for agricultural planning, resource allocation, and the decision-making process of the agricultural community.

5. CONCLUSION

This research introduced a novel approach for predicting rice yield, leveraging the ABC algorithm for feature selection along with a hybrid CNN+ LSTM model. The RF regressor complements the hybrid model by adding diversity and reducing prediction errors, exhibiting superior performance compared to other existing models by achieving the highest scores for important evaluation metrics. The visualization presented in the research further confirmed the model's ability to capture underlying patterns in the dataset. The practical implications of this research go beyond the academic sphere, providing significant benefits to the agricultural community. Accurate crop yield predictions empower farmers to optimize their practices, allocate resources effectively, and minimize crop losses. Policymakers can utilize these predictions for devising strategies related to food security, distribution management, and sustainable agriculture. Additionally, the commercial sector can make informed decisions regarding crop procurement, storage, and pricing based on this information. Although our hybrid model has shown impressive performance, there is still room for improvement. Future studies could focus on incorporating additional features like soil characteristics, historical climate data, and socio-economic factors. Leveraging GPUs to accelerate the model's training can help in faster predictive results. Evaluating the model on larger and more diverse datasets would also validate its robustness and generalizability across various regions and timeframes.

6. REFERENCES

[1] S. Keelery, "Annual yield of rice India FY 1991-2022", https://www.statista.com/statistics/764299/india-yield-of-rice/ (accessed:2023)

[2] P. Sathya, P. Gnanasekaran, “Paddy yield prediction in Tamil Nadu delta region using MLR-LSTM model”, Applied Artificial Intelligence, Vol. 37, No. 1, 2023.

[3] S. M. Bharath, S. Manoj, P. Adhappa, P. L. Patagar, R. Bhaskar, “Crop yield prediction with efficient use of fertilizers”, Lecture Notes in Electrical Engineering, Vol. 783, 2021, pp. 937-943.

[4] F. B. Felipe, L. H. A. Rodrigues, "The effect of tuning, feature engineering, and feature selection in data mining applied to rainfed sugarcane yield modeling”, Computers and electronics in agriculture, Vol. 128, 2016, pp. 67-76.

[5] C. Girish, F. Sahin, "A survey on feature selection methods", Computers & Electrical Engineering, Vol. 40, No. 1, 2014, pp. 16-28.

[6] Sánchez-Maroño, Noelia, A. Alonso-Betanzos, M. Tombilla-Sanromán, "Filter methods for feature selection--a comparative study", Lecture notes in Computer Science, Vol. 4881, 2007, pp. 178-187.

[7] N. El Aboudi, L. Benhlima, "Review on wrapper feature selection approaches", Proceedings of the IEEE International Conference on Engineering & MIS, Agadir, Morocco, 22-24 September 2016, pp. 1-5.

[8] P. S. M. Gopal, R. Bhargavi, “Performance evaluation of best feature subsets for crop yield prediction using machine learning algorithms”, Applied Artificial Intelligence, Vol. 33, No. 7, 2019, pp. 621-642.

[9] D. W. Christopher, J. M. Vance, H. K. Rasheed, A. Missaoui, K. M. Rasheed, F. W. Maier, "Using machine learning and feature selection for alfalfa yield prediction", AI, Vol. 2, No. 1, 2021, pp. 71-88.

[10] C. Gaurav, A. Chaudhary, "Crop recommendation system using machine learning algorithms", Proceedings of the IEEE 10th International Conference on System Modeling & Advancement in Research Trends, Greater Noida, India, 11-12 May 2021, pp. 109-112.

[11] S. V. Joshua et al. "Crop yield prediction using machine learning approaches on a wide spectrum", Computers, Materials & Continua, Vol. 72, No. 3, 2022, pp. 5663-5679.

[12] E. Banu, A. Geetha, “Rice crop yield prediction using random forest and deep neural network - an integrated approach”, SSRN Electron Journal, 2021.

[13] D. Elavarasan, P. M. Durairaj Vincent, “Crop yield prediction using deep reinforcement learning model for sustainable agrarian applications”, IEEE Access, Vol. 8, 2020, pp. 86886-86901.

[14] T. Islam, T. A. Chisty, A. Chakrabarty, "A Deep Neural Network Approach for Crop Selection and Yield Prediction in Bangladesh", Proceedings of the IEEE Region 10 Humanitarian Technology Conference, Malambe, Sri Lanka, 6-8 December 2018, pp. 1-6.

[15] A. Reyana, S. Kautish, P. M. S. Karthik, I. Ahmed Al-Baltah, M. B. Jasser, A. W. Mohamed, "Accelerating Crop Yield: Multisensor Data Fusion and Machine Learning for Agriculture Text Classification", IEEE Access, Vol. 11, 2023, pp. 20795-20805.

[16] H. Jing, H. Wang, Q. Dai, D. Han, "Analysis of NDVI data for crop identification and yield estimation", IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, Vol. 7, No. 11, 2014, pp. 4374-4384.

[17] A. Umair, S. A. Moqurrab, "Predicting crop diseases using data mining approaches: classification", Proceedings of the IEEE 1st International Conference on Power, Energy and Smart Grid, Mirpur Azad Kashmir, Pakistan, 9-10 April 2018, pp. 1-6.

[18] N. P. Sai, P. S. Venkat, B. L. Avinash, B. Jabber, "Crop yield prediction based on Indian agriculture using machine learning", Proceedings of the IEEE International Conference for Emerging Technology, Belgaum, India, 5-7 June 2020, pp. 1-4.

[19] R. Seireg, Hayam, Y. M. K. Omar, F. E. Abd El-Samie, A. S. El-Fishawy, A. Elmahalawy, "Ensemble machine learning techniques using computer simulation data for wild blueberry yield prediction", IEEE Access, Vol. 10, 2022, pp. 64671-64687.

[20] G. Yogesh, "A study on various data mining techniques for crop yield prediction", Proceedings of the IEEE International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques, Mysuru, India, 15-16 December 2017, pp. 420-423.

[21] I. E. Mladenova, J. D. Bolten, W. T. Crow, M. C. Anderson, C. R. Hain, D. M. Johnson, R. Mueller, "Intercomparison of soil moisture, evaporative stress, and vegetation indices for estimating corn and soybean yields over the US", IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, Vol. 10, No. 4, 2017, pp. 1328-1343.

[22] A. Manjula, G. Narsimha, "XCYPF: A flexible and extensible framework for agricultural Crop Yield Prediction", Proceedings of the IEEE 9th International Conference on Intelligent Systems and Control, Coimbatore, India, 9-10 January 2015, pp. 1-5.

[23] N. Lobna, I. E. Okwuchi, M. Saad, F. Karray, K. Ponnambalam, P. Agrawal, "Prediction of strawberry yield and farm price utilizing deep learning", Proceedings of the IEEE International Joint Conference on Neural Networks, Glasgow, UK, 19-24 July 2020, pp. 1-7.

[24] S. Pudumalar, E. Ramanujam, R. H. Rajashree, C. Kavya, T. Kiruthika, J. Nisha, "Crop recommendation system for precision agriculture", Proceedings of the IEEE Eighth International Conference on Advanced Computing, Chennai, India, 19-21 January 2017, pp. 32-36.

[25] N. Gnanasankaran, E. Ramaraj, T. Manikumar, "An Intelligent Framework for Rice Yield Prediction using Machine Learning based Models", International Journal of Scientific Engineering and Research, Vol. 12, No. 1, 2021.

[26] A. Saxena, M. Dhadwal, M. Kowsigan, "Indian Crop Production: Prediction and Model Deployment Using ML and Streamlit", Turkish Journal of Physiotherapy and Rehabilitation, Vol. 32, No. 3, 2021, p. 3.

[27] K. S. Saravanan, V. Bhagavathiappan, "Relative temperature disparity and rice yield across seasons in Tamil Nadu", Journal of Agrometeorology, Vol. 6, No. 2, 2004, pp. 5-9.

[28] S. Mishra, P. Paygude, S. Chaudhary, S. Idate, "Use of data mining in crop yield prediction", Proceedings of the 2nd International Conference on Inventive Systems and Control, Coimbatore, India, 19-20 January 2018, pp. 796-802.

[29] N. Gandhi, O. Petkar, L. J. Armstrong, "Rice crop yield prediction using artificial neural networks", Proceedings of the IEEE Technological Innovations in ICT for Agriculture and Rural Development, Chennai, India, 15-16 July 2016, pp. 105-110.

[30] V. Sharma, M. Shukla, R. Mandal, "Crop Analysis and Seed Marketing using Regression and Association Rules of India", Proceedings of International Conference on Emerging Trends in Information Technology and Engineering, Vellore, India, 24-25 February 2020, pp. 1-5.

[31] C. M. Manasa, B. P. Palayyan, "An Efficient Crop Yield Prediction Framework Using Hybrid Machine Learning Model", Revue d'Intelligence Artificielle Journal, Vol. 37, No. 4, 2023, pp. 1057-1067.

[32] T. Jiliang, S. Alelyani, H. Liu, "Feature selection for classification: A review", Data classification: Algorithms and applications, CRC Press, 2014, pp. 37-64.

[33] D. Karaboga, C. Ozturk, "A novel clustering approach: artificial bee colony (ABC) algorithm". Applied Soft Computing Journal, Vol. 11, No. 1, 2011, pp. 652-657.

[34] B. Akay, D. Karaboga, "A modified artificial bee colony algorithm for real-parameter optimization", Information Sciences, Vol. 192, 2012, pp. 120-142.

[35] D. Karaboga, B. Akay, "A comparative study of artificial bee colony algorithm", Applied Mathematics and Computation, Vol. 214, No. 1, 2009, pp. 108-132.

[36] K. Lata, S. Khan, "Experimental analysis of machine learning algorithms based on agricultural dataset for improving crop yield prediction", International Journal of Engineering and Advanced Technology, Vol. 9, No. 1, 2020, pp. 3246-3251.

Comparison of Deep Convolutional Neural Network Architectures for Fruit Categorization

Original Scientific Paper

Johan Muliadi Kerta
Bina Nusantara University,
School of Computer Scince, Computer Science
Bandung, Indonesia
johan.kerta@binus.ac.id

Abdul Haris Rangkuti
Bina Nusantara University,
School of Computer Scince, Computer Science
Bandung, Indonesia
rangku2000@binus.ac.id

Jeremy Tantio
Bina Nusantara University,
School of Computer Scince, Computer Science
Bandung, Indonesia
Jeremy.tantio@binus.ac.id

Abstract *– In general, there are so many types of fruit images that it is difficult for humans to differentiate them based on their visual characteristics alone. This research focuses on identifying and recognizing images of fruit from 23 different classes or types. Fruit varieties consist of 13 apple classes, 1 orange class, and 9 tomato classes, totaling 15,987 images. Fruit image data were collected from various sources, including the internet, magazines, and direct capture with a digital camera. The process of identifying and recognizing fruit images involves the classification of fruit images using a deep learning algorithm. Several CNN models, which are derivatives of deep learning, are used to achieve high accuracy and robustness in recognizing various types of apples and tomatoes. To evaluate the performance of each model, the apple data were trained on a large and diverse set of apple images using several CNN models such as ResNet50V2, InceptionV3, InceptionResNetV2, VGG16, VGG19, MobileNetV2, and EfficientNet. Performance is assessed using metrics such as accuracy, precision, recall, and F1 score. To achieve optimal performance in the image recognition process, it consists of preprocessing strategies, data augmentation, feature extraction, and classification supported by optimization, all of which have a significant impact on increasing accuracy performance. Experimental results show that certain CNN model architectures outperform other model architectures in terms of time efficiency and accuracy in recognizing fruit types/classes. However, to get more optimal results regarding the performance of the CNN model architecture for fruit categorization, two optimizers will be used, namely Adam and Adagrad, and will be compared. Based on Adam's optimizer experiments, the EfficientNet model produces the highest average accuracy of up to 99%, followed using the VGG 16 and ResNet V2 50 models, which achieve 98% and 97% accuracy. Meanwhile, the use of the Adagrad optimizer with the VGG 16 model produces the highest average accuracy of up to 95%, followed using the VGG 19 and EfficientNet models, which achieve accuracy of up to 93% and 91%. Overall, this experiment produced very good accuracy because it produced an average of above 90%. However, there is still room for improvement in recognizing fruits of different shapes, textures, and colors.*

Keywords*:* *fruit, identification, recognition, CNN, categorization, apple, tomato*

Received: November 12, 2023; Received in revised form: February 15, 2024; Accepted: February 16, 2024

1. INTRODUCTION

There are numerous types of fruits, which makes it difficult for humans to distinguish between them solely based on their characteristics. Additionally, there is a lack of user knowledge in differentiating between types of fruits and vegetables among various horticultural products in agriculture fields, particularly apples and tomatoes. This lack of knowledge makes it challenging for users to easily identify and select apples and tomatoes that are readily available in the market and are easily consumable [1]. To address this issue, it is important to provide users with information that will help them easily identify and select apples and tomatoes that are readily consumable. In general, identifying fruit objects through images is very useful because there are many types of fruit that exist and can be carried out. Fruit categorization has benefited greatly from deep learning techniques [2]. Accurate and efficient fruit categorization is crucial in agriculture, quality control, and automated fruit sorting systems Traditional methods of grading and sorting fruit by humans are slow, labor-intensive, error-prone, and tedious [3]. Therefore, there is a need for intelligent fruit grading systems. To address the challenges posed by differences in fruit appearance, shape, size, and orientation, researchers have developed a deep learning-based fruit categorization system [4]. The findings of this study contribute to the advancement of fruit classification systems and have practical implications in various fields, such as agriculture, the food industry, and automated fruit sorting [5]. In other studies architectures leverage deep learning techniques to eliminate the need for hard-coding specific features related to a fruit's shape, color, or other attributes. This method has the potential to enhance the accuracy and efficiency of fruit categorization operations, enabling automated and reliable fruit quality evaluation [6]. In conclusion, deep learning-based fruit categorization systems have proven to be effective in accurately classifying different varieties of fruits. These systems offer advantages such as non-contact operation, improved efficiency, and reliable fruit quality evaluation. They have practical applications in agriculture, the food industry, and automated fruit sorting systems [7].

The goal of the study was to create a robust and reliable model capable of accurately classifying different varieties of fruits. To achieve this, various deep learning models were employed, including ResNet50V2, InceptionV3, InceptionResNetV2, VGG16, VGG19, and EfficientNet. These models have demonstrated exceptional performance in picture classification tasks and are known for their ability to capture nuanced features and patterns. The research also involved extensive testing of different optimizers. Optimizers play a crucial role in deep learning models as they determine how the model learns and updates its parameters during training. Different optimizers have been shown to yield varying performance in terms of accuracy, precision, recall, and F1-score. However, specific details about the optimizers used in the study are not provided in the given information.

2. RELATED STUDY

Various types of evaluations and analyses were conducted on various classification models to identify citrus fruit diseases. This paper discusses concepts related to image acquisition processes, digital image processing, feature extraction, and classification approaches. Each concept is discussed separately [8]. It is crucial to employ image annotation techniques that are fast, simple, and highly effective. This research focuses on the agricultural sector and implements automatic image annotation to classify the ripeness of oil palm fruit and to identify different types of fruit. This approach aids farmers in improving fruit classification methods and increasing their production [9]. The fruit industry faces a common challenge: the lack of an automated system for classifying dates. Recent advancements in machine learning techniques have opened new opportunities for automating fruit classification and sorting tasks, traditionally handled by human experts [10].

This study explores the performance of various deep learning models and the impact of different parameters on the accuracy and efficiency of fruit classification systems using convolutional neural networks (CNNs) with various approaches [11]. This article highlights the application of AI in the food industry, maximizing resource utilization by reducing human error. Artificial intelligence, coupled with data science, can enhance the quality of restaurants, cafes, online food delivery chains, hotels, and food outlets by increasing production using different pairing algorithms for sales prediction [12]. In conducting the experiments using a dataset comprising images of 30 different fruit classes. The researchers employed prominent deep learning architectures, such as VGG16 and ResNet50, as the foundation for their classification system. They evaluated the models' performance based on accuracy, precision, recall, and F1-score. Their findings yielded 86% and 85% accuracy from the public dataset and 99% and 98% accuracy from their custom dataset [13].

By utilizing the Fruit-360 dataset, we ensure the dataset's reliability, backed by the success of previous research using this dataset. This research emphasizes the application of AI in the food industry, recommending significant capital savings through resource optimization, including human error reduction. This experiment employed a GPU as the primary processing power, achieving 177x acceleration on training data and 175x on test data [14]. In another study, a wider variety of fruits were used. The experiment was conducted using 24 classes of fruit comprising 3,924 images. The authors preprocessed the data by applying augmentation techniques. They implemented CNN, which trained the data with a batch size of 16 and 100 epochs, resulting in 95.5% accuracy for their test [15]. The comparison research used various kinds of apples, such as Granny Smith, Braeburn, Golden Delicious, and Cripps Pink, and other fruits, such as mandarin, lemon, and orange. It indicated that the average accuracy values for training and test datasets were 100% and 73%, respectively [16]. The advantages of artificial intelligence, deep learning-based computer vision can support various agricultural activities can be carried out automatically with maximum precision, making smart agriculture a reality [17].

Computer vision techniques, together with the ability to acquire high-quality images using remote cameras, enable non-contact and efficient technology-based solutions in agriculture [18, 19]. In another study, sea buckthorn fruits were used to quickly identify the moisture content range by collecting images of the appearance and morphology changes during the drying process [20]. Machine learning approaches for fruit classification have been proposed in the past, but deep learning, with its improved recognition and classification capabilities, can be a powerful engine for producing actionable results [21]. The classification of fruits can be divided into classes for edible and non-edible fruits, which is an important aspect in the industry. For example, one research project classified four fruits (Banana, Papaya, Mango, and Guava) into three stages: raw, ripe, and overripe [22].

Another study used a Convolutional Neural Network (CNN) to identify and classify different varieties of peanuts. Based on the deep learning technology, this paper improved the deep convolutional neural network VGG16 and applied the improved VGG16 to the identification and classification task of 12 varieties of peanuts [23]. In yet another study, a 13-layer CNN was designed, and various data augmentation methods were used, such as image rotation, Gamma correction, and noise injection. The researchers also compared maximum pooling with average pooling and used stochastic gradient descent with momentum to train the CNN [24]. The comparison table with existing studies can be seen in Table 1.

Table 1. The Comparison of some table with existing studies

Ref.	Tittle	Fruit Image used Method	Accuracy
6	Computerized Classification of Fruits using Convolution Neural Network	This research detect disease in fruit using a digital basis. Early detection of disease protects against damage to the entire plant using CNN	90%
9	Enhancing Image Annotation Technique of Fruit Classification Using a Deep Learning Approach	This research is about automatic image annotation which is repeated to classify the ripeness of oil palm fruit and recognize fruit varieties with Yolo based on Deep learning	98.7%
13	Fruits Classification and Detection Application Using Deep Learning	This paper is for an automatic fruit classification and detection system that has been developed using deep learning algorithms, namely Yolo and ResnetV2 or VGG16	85% and 98%
14	Analysis of artificial intelligence-based image classification techniques	This research is about an artificial intelligence-based image classification system for quickly identifying vegetables and fruits by looking through the camera billing process.	93%
17	Fruit image classification model based on MobileNetV2 with deep transfer learning technique	This research requires an automatic system to classify various types of fruit without the help of human labor using the modified version of MobileNetV2	99%
21	Fruit Classification Using Deep Learning	In this research about the importance of fruit classification for people who have dietary needs including to help them choose the right fruit category on a digital basis Using CNN Model	94.3%

Table 1. provides information about the comparison of several research references related to fruit image classification, including the model/algorithm used and the resulting accuracy of the model used in each reference. In the proposed system there are four classified fruits, namely Banana, Papaya, Mango, and Guava which are divided into three stages, namely unripe, ripe, and overripe fruit using a Convolutional Neural Network [25]. The fruit research process involves several steps, including fruit classification methodology, pre-processing, and the implementation of fruit classification using appropriate software and hardware. Preprocessing includes background removal and segmentation techniques to extract fruit areas.

3. PROOSED METHOD

In this classification research, the workflow is done based on this research method diagram in Fig. 1.

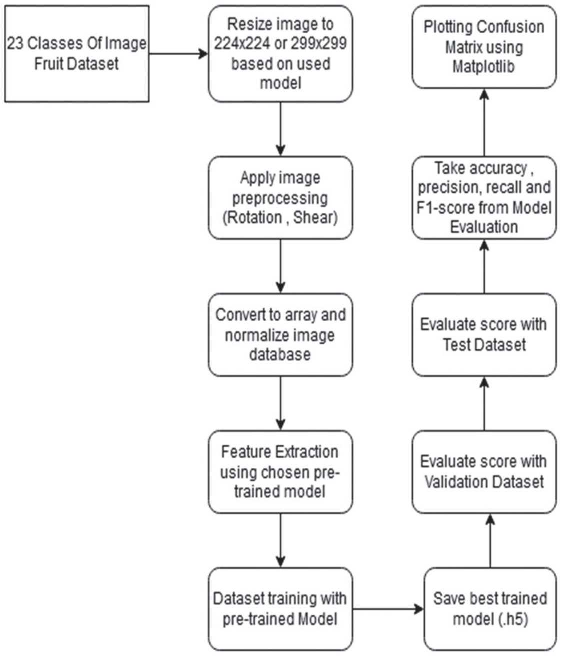

23 Classes Of Image Fruit Dataset
Resize image to 224x224 or 299x299 based on used model
Apply image preprocessing (Rotation , Shear)
Convert to array and normalize image database
Feature Extraction using chosen pre-trained model
Dataset training with pre-trained Model
Save best trained model (.h5)
Evaluate score with Validation Dataset
Evaluate score with Test Dataset
Take accuracy , precision, recall and F1-score from Model Evaluation
Plotting Confusion Matrix using Matplotlib

Fig. 1. Research diagram for "Fruit Image Classification Using Various Pre-Trained Models

The aim of experiment is to find the CNN model that yields optimal accuracy values. Two optimizer methods would be employed to support the classification performance in this experiment. To conduct this experiment, the images will undergo changes in pixel size and rotation. These modified images will then be converted into array values during the normalization process, which will be applied to all the images in the dataset. The array values would be processed to extract image characteristics based on the selected CNN model, both for training and testing data. A comparison process performed between the trained images and the testing images to determine the accuracy and performance of the classification. In summary, the stages involved in the classification of fruit images can be described as follows:

1. Data Preparation of Fruit Image Dataset

This research utilized the dataset obtained from Kaggle.com, specifically the Fruits-360 dataset. The dataset comprises 15,987 fruit images, covering 23 different fruit classes, with a primary focus on various types of apples and tomatoes. This diverse collection facilitates training and testing of CNN models, leading to improved classification accuracy. Additionally, these modifications have an impact on other accuracy measures during the experiment. An example of an apple can be seen in Fig. 2.

Fig. 2. Inform the Apple Braeburn class dataset which has undergone changes in rotation and focus

Meanwhile, the number of fruit classes in this study can be seen in Fig. 3.

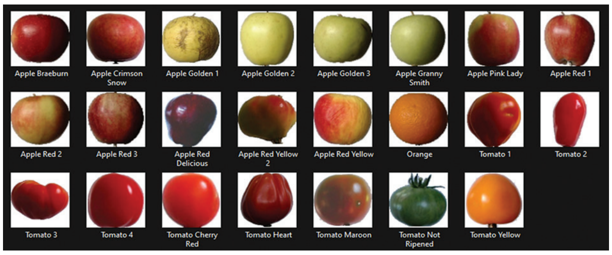


Fig. 3. Dataset Images from each Classes

Fig. 3 present the fruit dataset studied which consists of 13 classes of apples studied, 1 class of citrus fruit and 9 classes or types of tomatoes. They are apple Braeburn, Apple Crimson Snow, Apple Golden 1, Apple Golden 2, Apple Golden 3, Apple Granny Smith, Apple Pink Lady, Apple Red1, Apple Red2, Apple Red3, Apple Red Delicious, Apple Red Yellow, Apple Red Yellow2, Orange, Tomato1, Tomato2, Tomato3, Tomato cherry Red, Tomato Heart, Tomato Marcon, Tomato Not Ripened, Tomato Yellow.

2. Fruit Data Image Preprocessing (Resize, Rotation)

The fruit image data is pre-processed before being used for the dataset training process with a pre-trained model. Before, training the models, the fruit data images undergo preprocessing steps such as resizing and rotation. These steps help standardize the input images and ensure the models be able to handle variations in size and orientation. The following steps are taken for pre-processing the fruit image data:

1. Resizing: Fruit Image presents a full-frame fruit with different rotations and orientations. The dataset also includes resized fruit images with pixel sizes of 224 x 224 and 299 x 299. Each image is resized to either 299x299 or 224x224, depending on the model being used. The InceptionResNetV2, InceptionV3, and ResNet50V2 models require the image size to be 299x299, while the VGG16, VGG19, MobileNetV2, and EfficientNet models require the image size to be 224x224.
2. Rotation: After resizing the image, the entire dataset is reprocessed by applying different scale and shear rotations to each image. Moreover, each image undergoes rotation changes of 15 and 20 degrees, as well as shifts to ideal positions, enhancing recognition ease. These modifications aim to ensure equalized sizes within the fruit image dataset.
3. Selection of Pre-trained Models

To leverage the power of deep learning, pre-trained models are used in this study. Pre-trained models are neural network models that have been trained on large-scale datasets, such as ImageNet, and have learned to extract meaningful features from images. By using pre-trained models, researchers can benefit from the knowledge and representations learned by these models, saving time and computational resources.

3. Configuration and Optimization of Models

Configuring and optimizing the models involves setting up all the necessary parameters for processing the training data. This includes defining the architecture of the CNN models, specifying activation functions, kernel initializers, padding, input shape, and other relevant parameters. The models are then compiled and trained using the specified optimizers. The entire process of configuring and optimizing the models can be visualized through the output, which provides a comprehensive overview of the experiment.These resources are specifically tailored to configuring and optimizing the models for the task of fruit classification. This involves fine-tuning the models, adjusting hyperparameters, and selecting suitable optimization algorithms to achieve the best performance.

4. Training Dataset

The prepared data set is used to train the selected model. During the training process, the models learn to recognize and classify various types of fruits based on pre-prepared images and labels. The training data set is essential for the model to learn patterns and features that differentiate one class of fruit from another. For this training process, experiments have been carried out with several epochs of 10 and 50. As well as dividing the amount of fruit data used as training, testing and validation data.

5. Evaluation of Models on the Dataset

Once the models are trained, they are evaluated on a separate dataset to assess their performance. This evaluation dataset contains images that the models have not seen during training. By evaluating the models on unseen data, researchers can measure their generalization ability and determine how well they can classify fruits in real-world scenarios. The dataset consists of a total of 15,987 images with different rotations and orientations, including 9,600 training datasets, 2,386 validation datasets, and 4,001 testing datasets.

6. Comparative Analysis of CNN Models

A comparative analysis is conducted to compare the performance of the different CNN models used in this study. This analysis helps to identify the strengths and weaknesses of each model and provides insights into which models are most effective for fruit classification. Overall, this study focuses on the image processing of apples and tomatoes, utilizing deep learning algorithms and optimizations to achieve optimal classification accuracy. The experimental implementation is divided into several stages, including data preparation, image preprocessing, model selection, configuration and optimization, training, evaluation, and comparative analysis of CNN models.

4. EXPERIMENTAL RESULTS

In this experiment on fruit image classification, the model trained the image as many as 10 and 50 epochs of the given test time. However, it based on the test results on the fruit images obtained the accuracy results of the tests that have been carried out quite satisfactory and convincing, as well as being able to draw conclusions regarding the success of this research. The result of training showed with Matplotlib for better understanding of each training process. All the results shown in Fig. 4.

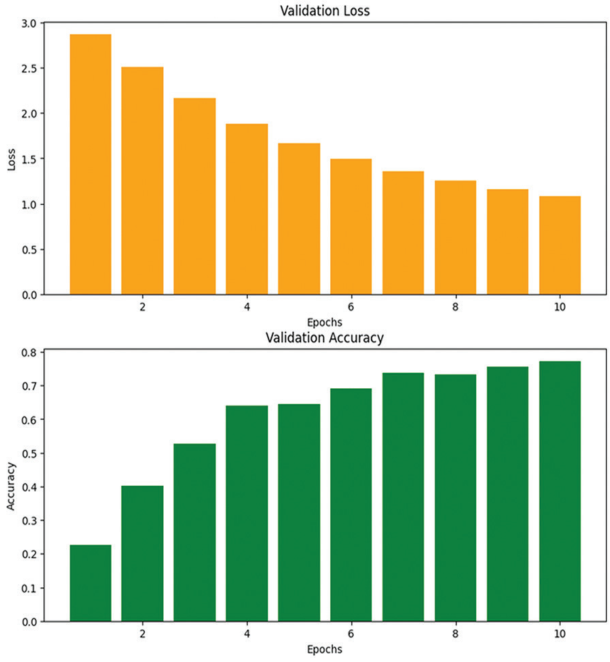


Fig. 4. Validation Dataset Accuracy and Loss on InceptionV3 through 10 Epochs using Adagrad optimizer

In Fig. 4, the experiment focused on comparing the performance of the Adagrad optimizer. The initial accuracy of the Adagrad optimizer was not as high as that of the Adam optimizer. However, the experiment showed that the Adagrad optimizer consistently improved over time throughout the epochs. Although it did not achieve the same level of accuracy as Adam in this test.

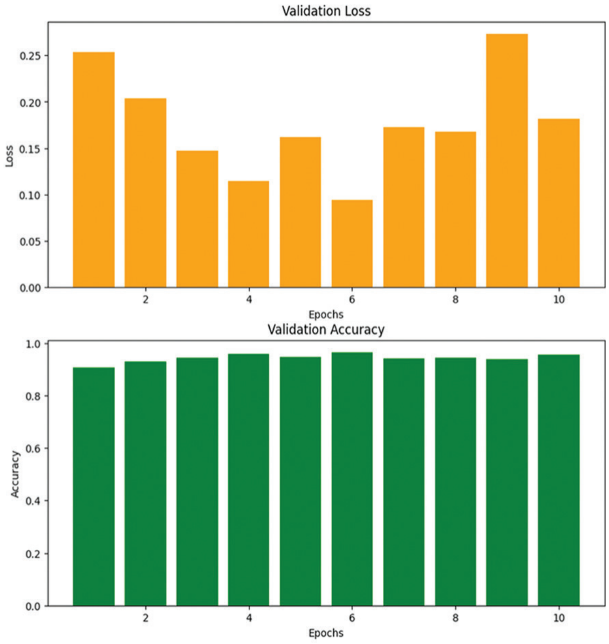
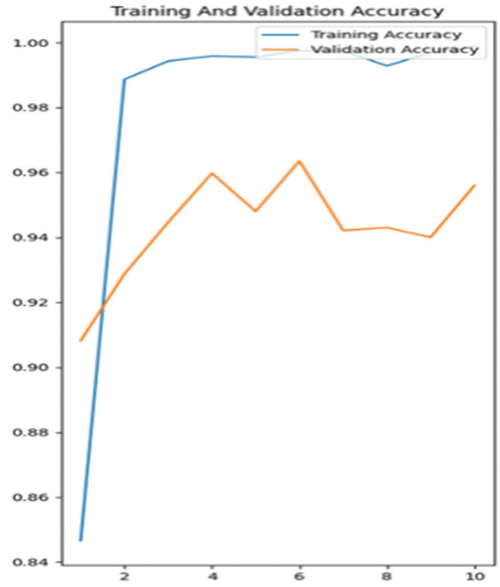
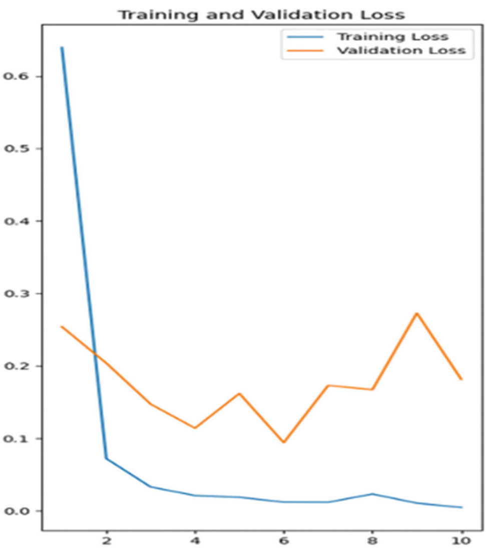


Fig. 5. Validation Dataset Accuracy and Loss on InceptionV3 through 10 Epochs using Adam optimizer

In Fig. 5 inform about the experiment of image classification used Adam optimizer and then the accuracy from the beginning was already high but the accuracy and losses seems to fluctuate a lot although the difference wasn't much with the numbers fluctuate around 0,1 between the fluctuation.



a) Training and Validation Accuracy Curves using Adam Optimizer on ResNet50V2

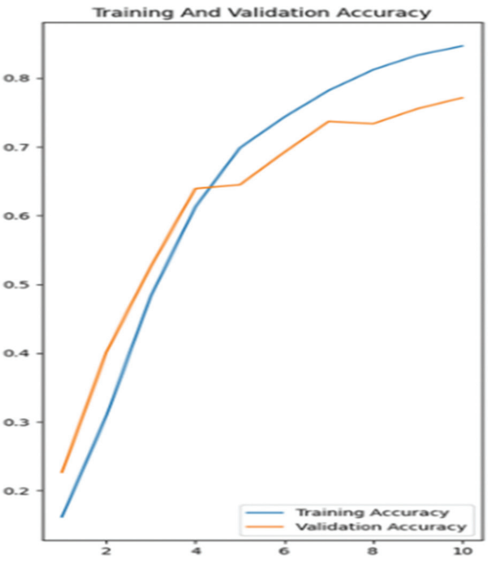


(b) Training and Validation Loss Curves using Adam Optimizer on ResNet50V2

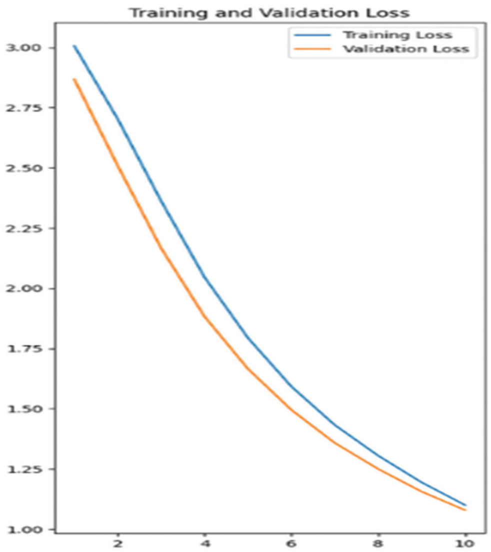
Fig. 6. Training and Validation Accuracy include Loss Curves using Adam Optimizer on ResNet50V2

In Fig. 6 present the training and validation accuracy dataset, as well as the increase in loss during training. The experiment used ResNet50V2 and Adam optimizer shows high accuracy up to an average of 90%, indicating good fluctuations in the training process.

In Fig. 7 present the training and validation accuracy include loss process used Adagrad optimizer starts with an accuracy more of 86% but shows consistency during training. In-depth analysis, the experiment extends the model training to 50 epochs. In this experiment, the EfficientNet model was used with the Adam optimizer, achieving an average accuracy of 97%. When using the Adagrad optimizer, the accuracy reached 92%. Additionally, the training loss with the Adam optimizer was smaller than with Adagrad.



(a) Training and Validation Accuracy Curves using Adagard Optimizer on ResNet50V2



(b) Training and Validation Loss Curves using Adagard Optimizer on ResNet50V2

Fig. 7. Training and Validation Accuracy include Loss Curves using Adagard Optimizer on ResNet50V2

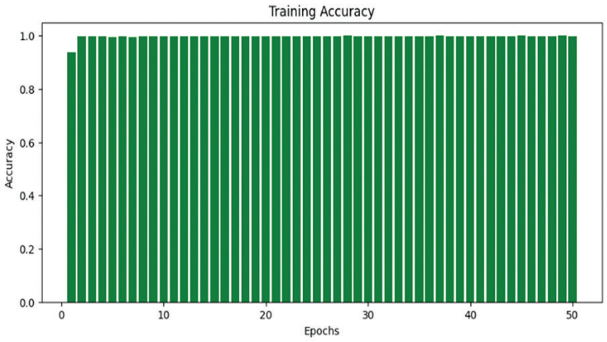


Fig. 8. Training accuracy used EfficientNet model and Adam Optimizer

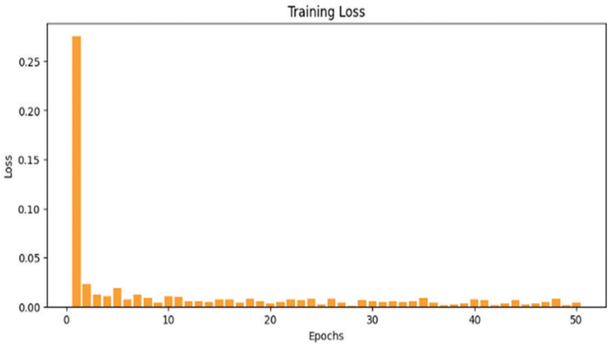


Fig. 9. Training loss used EfficientNet model and Adam Optimizer

Fig. 8 and Fig. 9 present the training accuracy and loss process results using the EfficientNet model by carrying out a 50-iteration. The image obtained shows that the accuracy using the Adam optimizer has an accuracy of almost 100% and however, Fig. 10 and Fig. 11 present the training accuracy and loss process results used Adagard optimizer which had an accuracy of 80 - 90% results. In this experiment present confusion matrices for all models on the test dataset to provide a detailed of model performance.

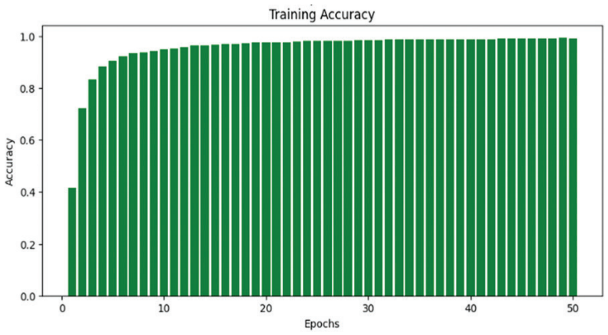


Fig. 10. Training accuracy used EfficientNet model and Adagard Optimizer

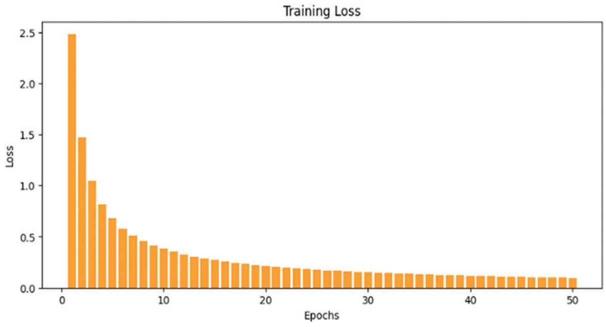
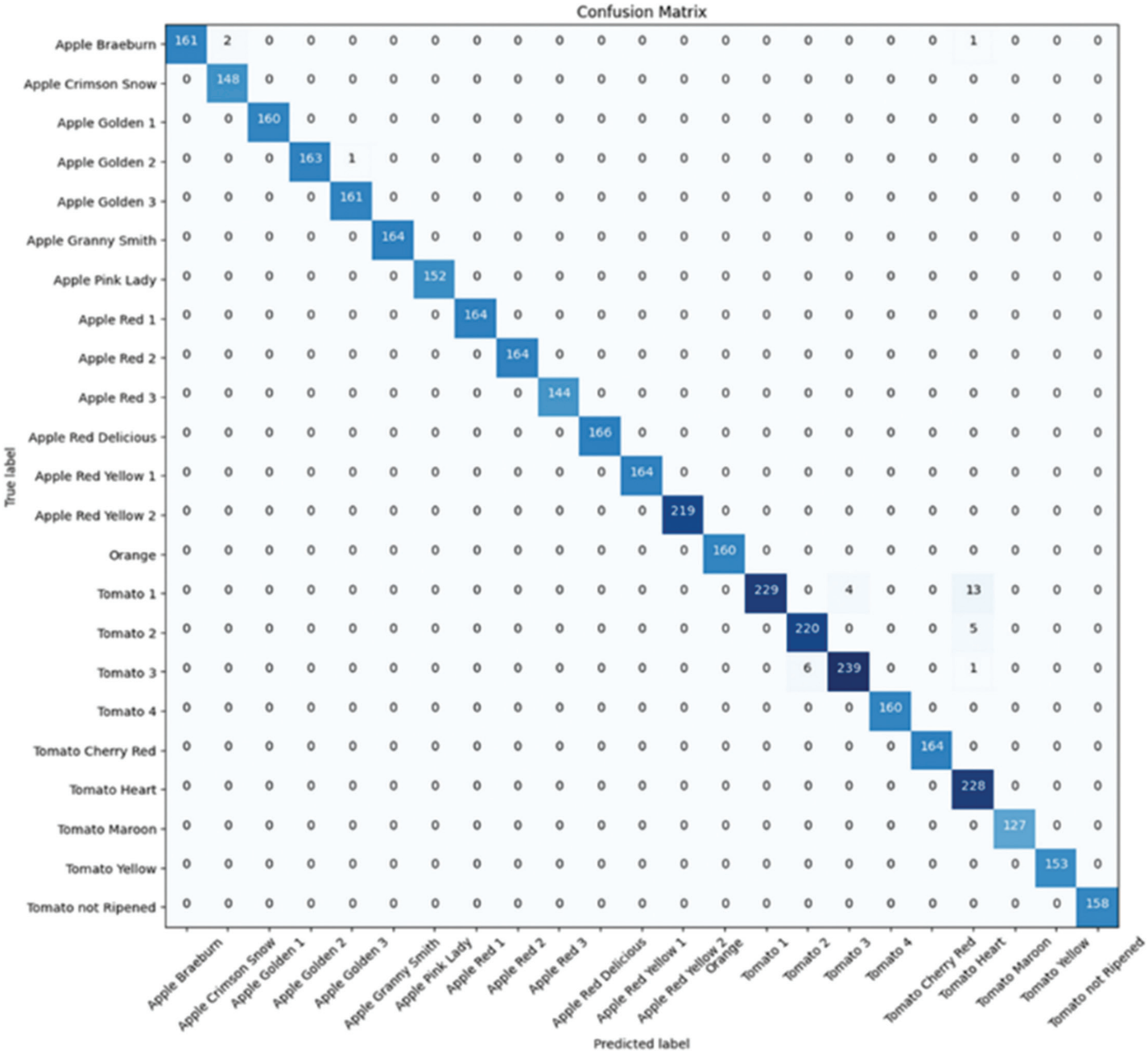


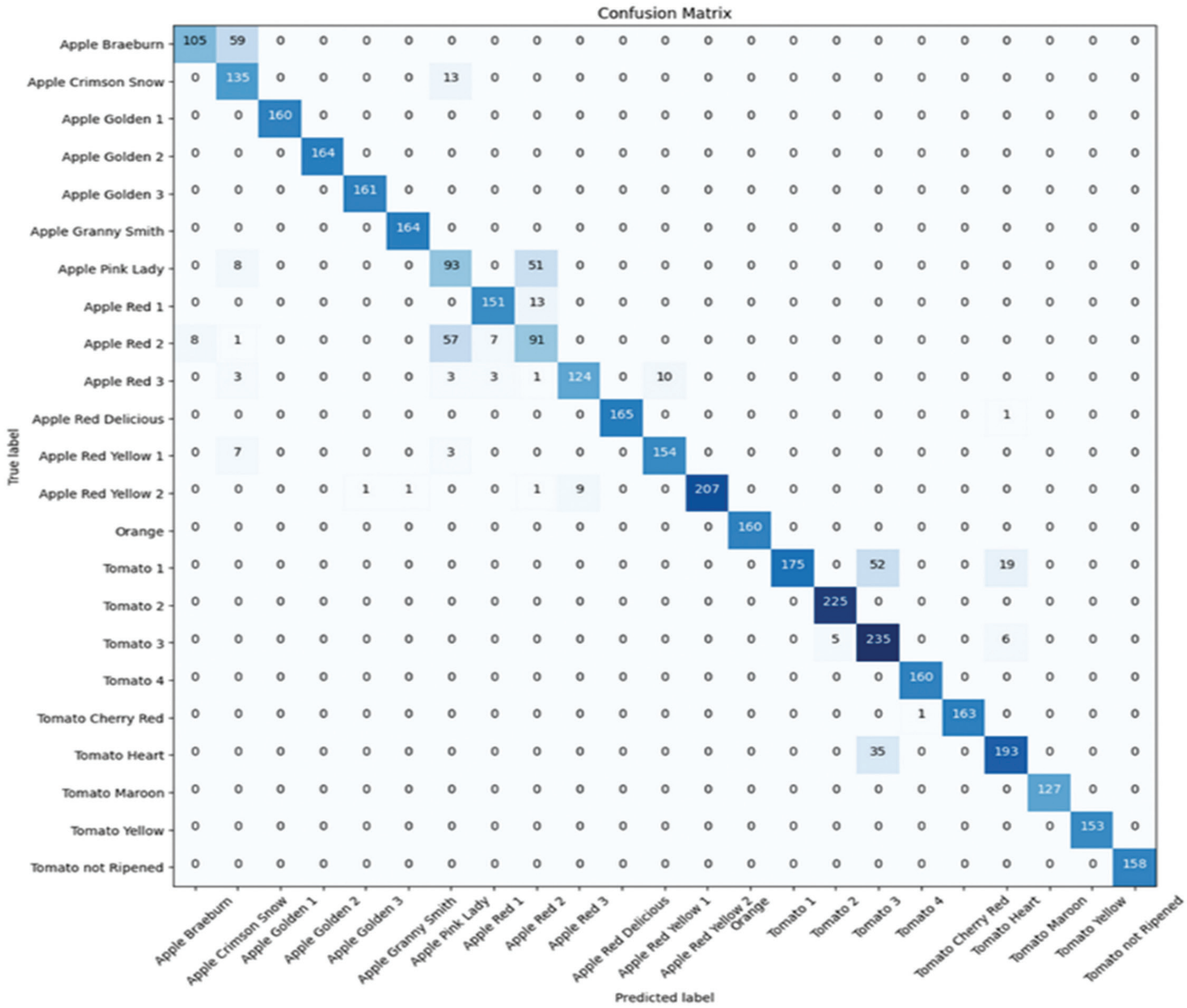
Fig. 11. Training loss used EfficientNet model and Adagard Optimizer

In Fig. 12 and Fig. 13 inform the confusion matrix graph shows that the overall performance evaluation in classifying fruit image data shows that the prediction results of the CNN model with the Adam optimizer produce better accuracy than Adagard. In this experiment, the EfficientNet model was used which produced prediction performance accuracy between 98 - 100%. whereas with Adagard only a few of fruits images can be detected properly.



	Apple Braeburn	Apple Crimson Snow	Apple Golden 1	Apple Golden 2	Apple Golden 3	Apple Granny Smith	Apple Pink Lady	Apple Red 1	Apple Red 2	Apple Red 3	Apple Red Delicious	Apple Red Yellow 1	Apple Red Yellow 2	Orange	Tomato 1	Tomato 2	Tomato 3	Tomato 4	Tomato Cherry Red	Tomato Heart	Tomato Maroon	Tomato Yellow	Tomato not Ripened
Apple Braeburn	161	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Apple Crimson Snow	0	148	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Apple Golden 1	0	0	160	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Apple Golden 2	0	0	0	163	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Apple Golden 3	0	0	0	0	161	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Apple Granny Smith	0	0	0	0	0	164	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Apple Pink Lady	0	0	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Apple Red 1	0	0	0	0	0	0	0	164	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Apple Red 2	0	0	0	0	0	0	0	0	164	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Apple Red 3	0	0	0	0	0	0	0	0	0	144	0	0	0	0	0	0	0	0	0	0	0	0	0
Apple Red Delicious	0	0	0	0	0	0	0	0	0	0	166	0	0	0	0	0	0	0	0	0	0	0	0
Apple Red Yellow 1	0	0	0	0	0	0	0	0	0	0	0	164	0	0	0	0	0	0	0	0	0	0	0
Apple Red Yellow 2	0	0	0	0	0	0	0	0	0	0	0	0	219	0	0	0	0	0	0	0	0	0	0
Orange	0	0	0	0	0	0	0	0	0	0	0	0	0	160	0	0	0	0	0	0	0	0	0
Tomato 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	229	0	4	0	0	13	0	0	0
Tomato 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	220	0	0	0	5	0	0	0
Tomato 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	239	0	0	1	0	0	0
Tomato 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	160	0	0	0	0	0
Tomato Cherry Red	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	164	0	0	0	0
Tomato Heart	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	228	0	0	0
Tomato Maroon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	127	0	0
Tomato Yellow	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	153	0
Tomato not Ripened	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	158

Fig.12. The confusion matrices used EfficientNet model and Adam Optimizer



	Apple Braeburn	Apple Crimson Snow	Apple Golden 1	Apple Golden 2	Apple Golden 3	Apple Granny Smith	Apple Pink Lady	Apple Red 1	Apple Red 2	Apple Red 3	Apple Red Delicious	Apple Red Yellow 1	Apple Red Yellow 2	Orange	Tomato 1	Tomato 2	Tomato 3	Tomato 4	Tomato Cherry Red	Tomato Heart	Tomato Maroon	Tomato Yellow	Tomato not Ripened
Apple Braeburn	105	59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Apple Crimson Snow	0	135	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Apple Golden 1	0	0	160	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Apple Golden 2	0	0	0	164	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Apple Golden 3	0	0	0	0	161	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Apple Granny Smith	0	0	0	0	0	164	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Apple Pink Lady	0	8	0	0	0	0	93	0	51	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Apple Red 1	0	0	0	0	0	0	0	151	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Apple Red 2	8	1	0	0	0	0	57	7	91	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Apple Red 3	0	3	0	0	0	0	3	3	1	124	0	10	0	0	0	0	0	0	0	0	0	0	0
Apple Red Delicious	0	0	0	0	0	0	0	0	0	0	165	0	0	0	0	0	0	0	0	1	0	0	0
Apple Red Yellow 1	0	7	0	0	0	0	3	0	0	0	0	154	0	0	0	0	0	0	0	0	0	0	0
Apple Red Yellow 2	0	0	0	0	1	1	0	0	1	9	0	0	207	0	0	0	0	0	0	0	0	0	0
Orange	0	0	0	0	0	0	0	0	0	0	0	0	0	160	0	0	0	0	0	0	0	0	0
Tomato 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	175	0	52	0	0	19	0	0	0
Tomato 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	225	0	0	0	0	0	0	0
Tomato 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	235	0	0	6	0	0	0
Tomato 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	160	0	0	0	0	0
Tomato Cherry Red	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	163	0	0	0	0
Tomato Heart	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	0	0	193	0	0	0
Tomato Maroon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	127	0	0
Tomato Yellow	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	153	0
Tomato not Ripened	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	158

Fig.13. The confusion matrices used EfficientNet model and Adagard Optimizer

After doing the training and testing the fruit image dataset, the next step is to evaluate and test the model using test dataset for every model which supported with Optimizers. In this section, we presented the results of our tests using 7 pre-trained models and 2 optimizers. It is important to mention that we have tested 1 Orange dataset, which achieved 100% accuracy, so we will not provide further details about it in this description.

a) InceptionV3

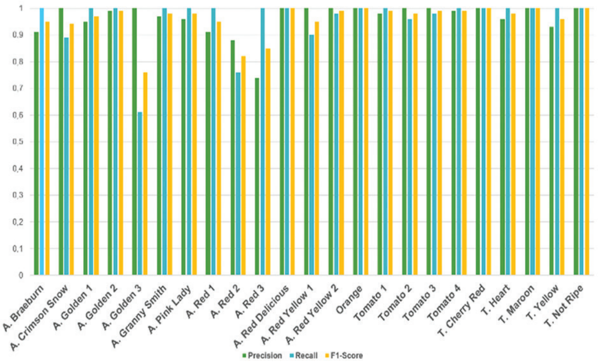


Fig. 14. Accuracy Classification using InceptionV3 and the Adam optimizer

Fig. 14 shows the results of experiments using InceptionV3 and the Adam optimizer. The classes of tomatoes, namely Red Delicious, Orange, Cherry Red, Maroon, and Note Ripe, achieved an accuracy of 100%. Additionally, the average accuracy of this experiment reached 96%. The experiment also yielded accuracy results of 100% for several other tomato classes.

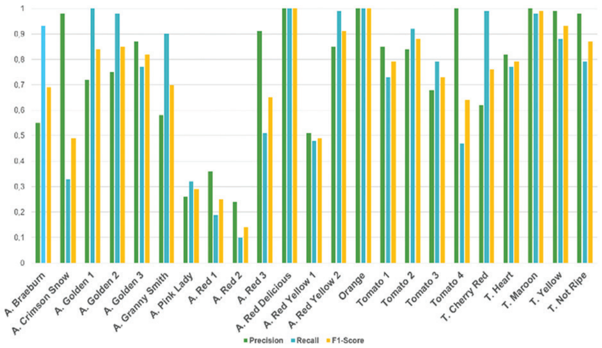


Fig. 15. Accuracy Classification using InceptionV3 and the Adagrad optimizer

In Fig. 15, the experiment results are shown using the InceptionV3 model and the Adagrad optimizer. It is observed that Orange and A. Red Delicious achieved 100% accuracy.

The dataset experiment overall achieved an average accuracy of 74%. However, the accuracy results for Apple Red 1 and 2, including pink lady, were significantly lower with an average accuracy of only 16% in this experiment.

b) InceptionResNetV2

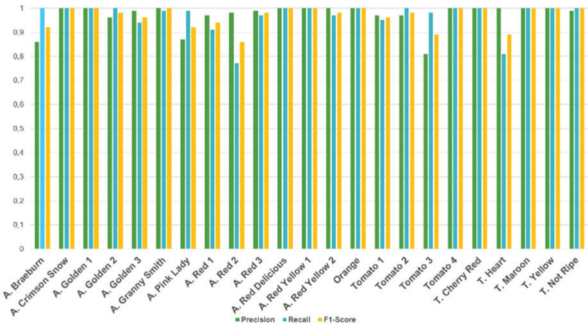


Fig.16. Accuracy Classification using InceptionResNetV2 and the Adam optimizer

In Fig. 16 the results of the experiment using the InceptionResNetV2 and Adam optimizer are reported. The following fruits achieved 100% accuracy: Crimson Snow, Golden 1, Red Delicious, Red Yellow1, Tomato 4, Cherry Red, Maroon, Yellow, and Not Ripe. The dataset experiment achieved an average accuracy of 74%. However, when using this CNN model, 10 types of fruit can be recognized with 100% accuracy, while the accuracy for other types of fruit can reach up to 80%.

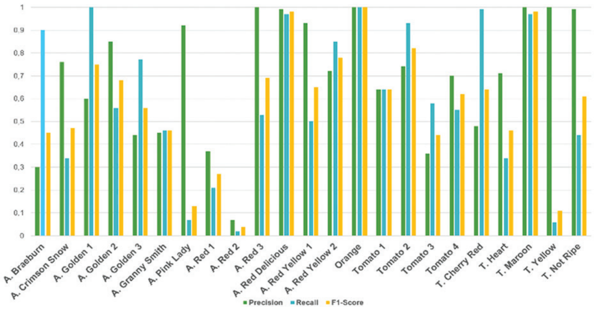


Fig.17. Accuracy Classification using InceptionResNetV2 and the Adagrad optimizer

In Fig. 17, the results of the experiment using the InceptionResnetV2 model and the Adagrad optimizer are presented. It is observed that the Orange class achieved 100% accuracy. Other types of fruits, such as Red Delicious and Maroon, achieved an accuracy rate above 95%. However, in this experiment, both Pink Lady and Red 2 showed low accuracy, with an average of less than 10%.

c) ResNet50V2

In Fig.18, the results of the experiment using the ResNetV2-50 model and the Adam optimizer are presented. According to this experiment, around 12 fruits achieved 100% accuracy, including Golden 1 and 2, Red Yellow, Orange, Red Delicious, and others. The dataset experiment achieved an average accuracy of 88% using this model.

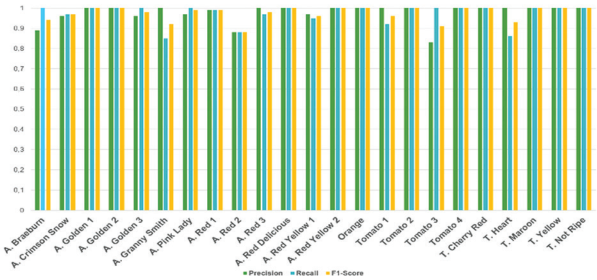


Fig. 18. Accuracy Classification ResNet50V2 using the Adam optimizer

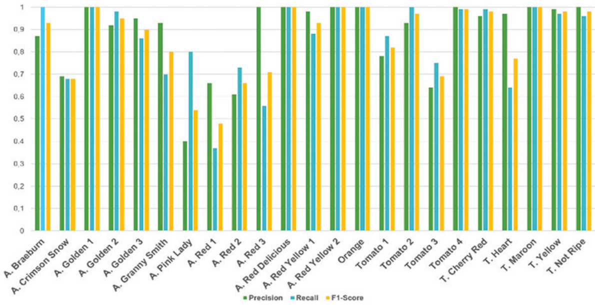


Fig. 19. Accuracy Classification using ResNetV2 and the Adagrad optimizer

In Fig. 19 present the experiment results using the ResNetV2-50 model and the Adagrad optimizer show that Orange and A. Red Delicious have 100% accuracy. The dataset experiment achieved an average accuracy of 74%. However, the accuracy results for Apple Red 1 and 2, including Pink Lady, were low, with an average accuracy of only 16% in this experiment.

d) VGG16

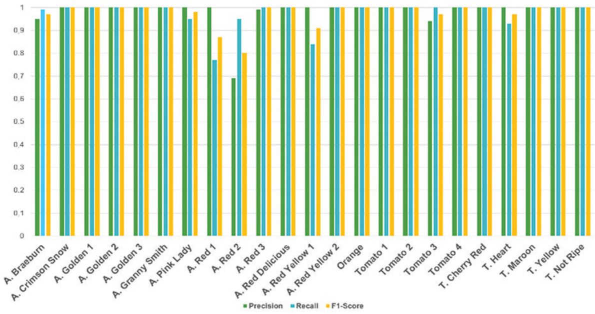


Fig. 20. Accuracy Classification on VGG16 using the Adam optimizer

In Fig. 20 present the experiment result using the VGG 16 and the Adam optimizer. In this experiment around 15 Fruits have achieved 100% accuracy such as A. Golden 1 and 2, A. Red Yellow, Orange, Tomato and A. Red Delicious and others. The experiment achieved the average more than 94% accuracy using this model. Based on this result can be concluded using this model has the best accuracy results.

In Fig. 21 present the experimental results using the VGG16 model and the Adagrad optimizer. In this experiment, around 10 fruits have achieved 100% accuracy, including Golden 1, Red Delicious, Red Yellow2, Orange, Tomato1, and other tomato varieties. The dataset experiment achieved an average accuracy of over 87% using this model. The lowest accuracy observed with the VGG16 was approximately 60%.

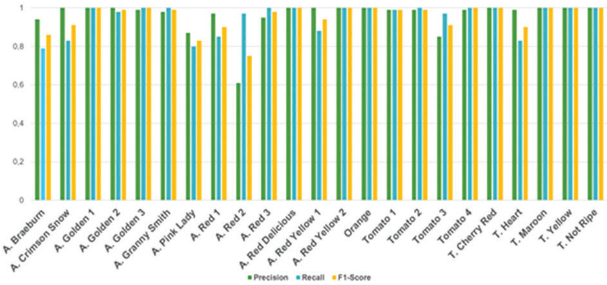


Fig. 21. Accuracy Classification using VGG16 and the Adagrad optimizer

e) MobileNetV2

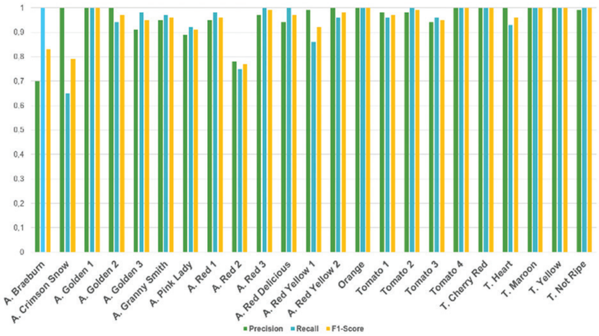


Fig. 22. Accuracy Classification using MobileNetV2 and the Adam optimizer

In Fig. 22 present the experiment results using the MobileNetV2 model and the Adam optimizer are presented. In this experiment, around 6 fruits have achieved 100% accuracy, including Golden 1 Orange, as well as various other kinds of tomatoes. The dataset experiment has achieved an average accuracy of over 82% using this model. The lowest accuracy achieved using the MobileNetV2 model was approximately 68%.

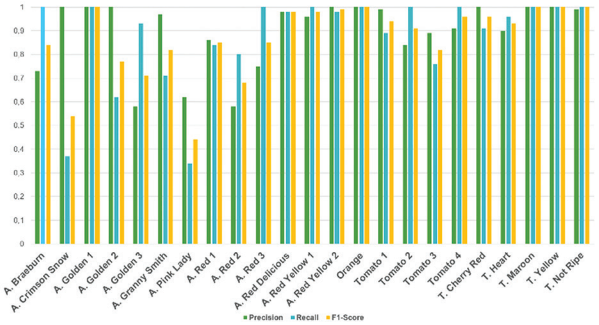


Fig. 23. Accuracy Classification on MobileNetV2 using the Adagrad optimizer

In Figure 23, the experiment results using the MobileNetV2 model and Adam optimizer are presented. In this experiment, around 5 fruits achieved 100% accuracy, including Golden 1 Orange and various types of tomatoes. The dataset experiment achieved an average accuracy of over 82% using this model. The lowest accuracy was observed with the MobileNetV2 model, which was about 64%.

f) EfficientNet

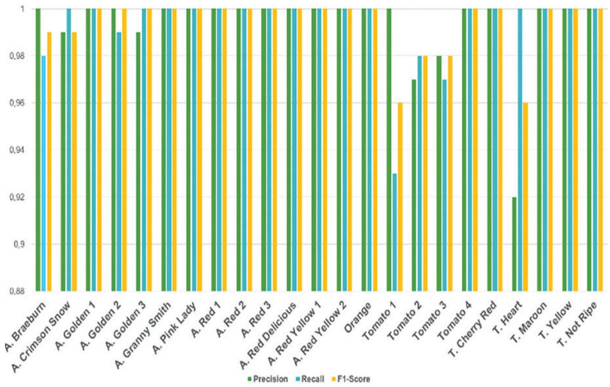


Fig. 24. Accuracy Classification on EfficientNet using the Adam optimizer

In Fig. 24, the experiment results using the EfficientNet and Adam optimizer are presented. In this experiment, around 15 fruits achieved 100% accuracy, including A. Golden 1, A. Granny Smith, A. Pink Lady, and apple varieties, as well as tomato varieties. The experiment achieved an average accuracy of more than 92% using this model. EfficientNet and Adam Optimizer emerged as the most successful among the 7 models tested with Adam optimizer. Interestingly, EfficientNet exhibited a different behaviour compared to the other models. While the models struggled to identify Apple Red, EfficientNet had the lowest accuracy in identifying Tomato Heart, but it still achieved 90% accuracy.

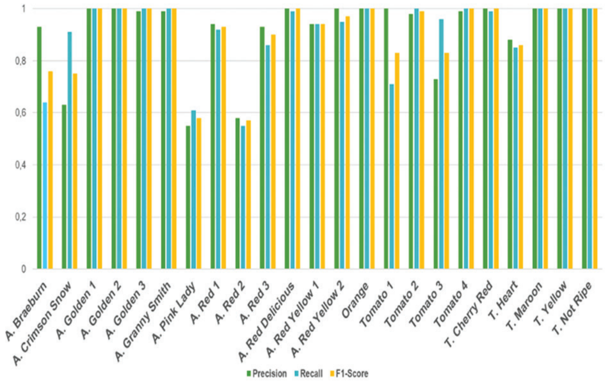


Fig. 25. Accuracy Classification on EfficientNet using the Adagrad optimizer

In Fig. 25 present the experiment results using the EfficientNet and Adagrad optimizer are presented. In this experiment, approximately six fruits achieved 100% accuracy, including A. Golden 1, A. Golden 2, Orange, and other types of tomato fruits. The experiment achieved an average accuracy of 73% using this model.

5. EVALUATION AND DISCUSSION

Based on the results from the testing dataset, we have concluded that the experiment conducted using the Adam Optimizer performed better than the one using the Adagrad Optimizer. When conducting the experiment with the Adam Optimizer, the image fruit dataset achieved an average accuracy of approximately 96.85% with a loss of only 0.85%. On the other hand, when the Adagrad optimizer was used, the dataset had an average accuracy of about 85.5% with a significantly higher average loss of 60%. The results of all the experiments are presented in Figs. 22 and 23.

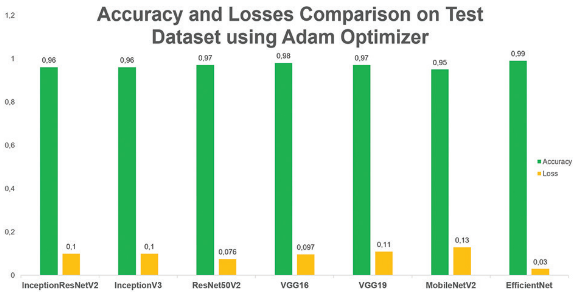


Fig. 26. Average Results of Accuracy and Losses after Testing Using Test Dataset and the Adam Optimizer

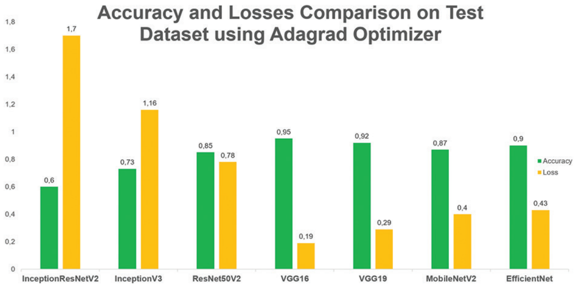


Fig. 27. Average Model Accuracy and Losses after testing using Test Dataset using the Adagrad Optimizer

In Fig. 26 dan Fig. 27 present on several experiments on recognizing types of fruit that have been carried out using several CNN models and optimizers. The results can be seen in Fig. 24 and Fig. 25.

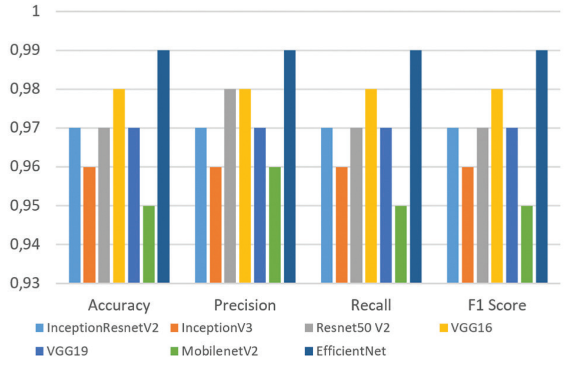


Fig. 28. Average accuracy results, precision, recall, and F1 Score, on several CNN models supported by Adam Optimizer

Fig. 28 presents the highest accuracy, precision, recall and F1 Score results with several CNN models in banana fruit classification with the EfficientNet, VGG 16, and VGG 19 models supported by Adam Optimizers with performance of 99%, 98% and 97%. However, if you experiment using the Adagard optimizer can be seen in Fig. 25.

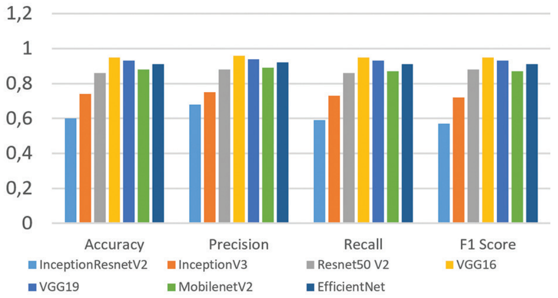


Fig. 29. Average accuracy results, precision, recall, and F1 Score, on several CNN models supported by Adagard Optimizer

Fig. 29 presents the highest accuracy, precision, recall and F1 Score results with several CNN models in banana fruit classification with the VGG 16, VGG 19 and EfficientNet models supported by Adam Optimizers with performance of 95%, 93% and 91%.

In recognition process is optimal because in the extracting image characteristics, it is divided into 3 values height, weight, and dimension. The experimental process using 2 optimizers and 7 CNN models. The conclusion got from the experiment using this dataset where The Adam optimizer is better when it comes to training and classifying fruit image dataset. Adagrad optimizer does not perform well in the model accuracy in such small epochs used but from our observations, Adagrad have a good consistency when it comes to training model. Meanwhile Adam optimizer already have a good accuracy starting from early epochs, but the accuracy and losses fluctuate a lot, so it creates inconsistencies. However, Adagrad in the other hand have a bad accuracy and losses in such small epochs but showed steady improvement over training. Extended research using the shape characteristic is needed to prove the hypothesis of this theory.

6. CONCLUSIONS

In conclusion, this research used the fruit dataset and made several key findings:

1. The feature extraction process in this research involved using a library to process images with three dimensions: height, width, and channel. Additionally, a value of 1 was added to indicate whether the elaboration process was completed for each image.
2. The research utilized CNN algorithms, employing seven different models: ResNet50V2, InceptionResNetV2, InceptionV3, VGG16, VGG19, MobileNetV2, and EfficientNet.

3. Among these models, VGG16 demonstrated the best performance, achieving 98% accuracy with the Adam optimizer and 95% accuracy with the Adagrad optimizer.

4. The Adam optimizer proved to be a superior option for fruit classification research. In contrast, the Adagrad optimizer resulted in poor accuracy and high losses during training, which negatively impacted the experiment's outcome. It is worth noting that the number of epochs used may have influenced Adagrad's poor performance.

5. Notably, the Apple Red 2 dataset consistently exhibited the lowest accuracy across all tests. This discrepancy may be attributed to the fact that apples themselves can have different colors depending on their orientation.

7. REFERENCES:

[1] M. K. Tripathi, D. D. Maktedar, "A role of computer vision in fruits and vegetables among various horticulture products of agriculture fields: A survey", Information Processing in Agriculture, Vol. 7, No. 2, 2020, pp. 183-203.

[2] R. Khan, R. Debnath, "Multi class fruit classification using efficient object detection and recognition techniques", International Journal of Image, Graphics and Signal Processing, Vol. 11, No. 1, 2019.

[3] B. Zhang, B. Gu, G. Tian, J. Zhou, J. Huang, Y. Xiong, "Challenges and solutions of optical-based nondestructive quality inspection for robotic fruit and vegetable grading systems: A technical review", Trends in Food Science & Technology, Vol. 81, 2018, pp. 213-231.

[4] Seema, A. Kumar, G. S. Gill, "Automatic fruit grading and classification system using computer vision: A review", Proceedings of the Second International Conference on Advances in Computing and Communication Engineering, Dehradun, India, 1-2 May 2015, pp. 598-603.

[5] M. K. Tripathi, D. D. Maktedar. "A role of computer vision in fruits and vegetables among various horticulture products of agriculture fields: A survey", Information Processing in Agriculture, Vol.7, No. 2, 2020, pp. 183-203.

[6] R. Yamparala, R. Challa, V. Kantharao, P. S. R. Krishna, "Computerized classifcation of fruits using convolution neural network", Proceedings of the International Conference on Smart Structures and Systems, Chennai, India, 23-24 July 2020.

[7] K. Hameed, D. Chai, A. Rassau, "A comprehensive review of fruit and vegetable classification techniques", Image and Vision Computing, Vol. 80, 2018, pp. 24-44.

[8] P. Dhiman, A. Kaur, V. R. Balasaraswathi, Y. Gulzar, A. A. Alwan, Y. Hamid, "Image Acquisition, Preprocessing and Classification of Citrus Fruit Diseases: A Systematic Literature Review", Sustainability, Vol. 15, No. 12, 2023, p. 9643.

[9] N. Mamat, M. F. Othman, R. Abdulghafor, A. A. Alwan, Y. Gulzear, "Enhancing image annotation technique of fruit classification using a deep learning approach", Sustainability, Vol. 15, No. 2, 2023, p. 901.

[10] K. Bresilla, G. D. Perulli, A. Boini, B. Morandi, L. C. Grappadelli, L. Manfrini. "Single-shot convolution neural networks for real-time fruit detection within the tree", Frontiers in Plant Science, Vol. 10, 2019, p. 611.

[11] R. Yacouby, D. Axman, "Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models", Proceedings of the first workshop on evaluation and comparison of NLP systems, November 2020, pp. 79-91. (online)

[12] K. Albarrak, Y. Gulzar, Y. Hamid, A. Mehmood, A. B. Soomro, "A Deep Learning-Based Model for Date Fruit Classification", Sustainability, Vol. 14, No. 10, 2022, p. 6339.

[13] N. E. Mimma, S. Ahmed, T. Rahman, R. Khan, "Fruits Classification and Detection Application Using Deep Learning", Scientific Programming, Vol. 2022, 2022.

[14] S. Shakya, "Analysis of artificial intelligence-based image classification techniques", Journal of Innovative Image Processing, Vol. 2, No. 1, 2020, pp. 44-54.

[15] I. Kumar, J. Rawat, N. Mohd, S. Husain, "Opportunities of artificial intelligence and machine learning in the food industry", Journal of Food Quality, Vol. 2021, 2021.

[16] A. Nasiri, A. Taheri-Garavand, Y. D. Zhang, "Image-based deep learning automated sorting of date

fruit", Postharvest Biology and Technology, Vol. 153, 2019, pp. 133-141.

[17] Y. Gulzar, "Fruit image classification model based on MobileNetV2 with deep transfer learning technique", Sustainability, Vol. 15, No. 3, 2023, p. 1906.

[18] V. G. D. Dhanya, A. Subeesh, N. L. Kushwaha, D. K. Vishwakarma, T. N. Kumar, G. Ritika, A. N. Singh, "Deep learning-based computer vision approaches for smart agricultural applications", Artificial Intelligence in Agriculture, Vol. 6, 2022, pp. 211-229.

[19] J. Naranjo-Torres, M. Mora, R. Hernández-García, R. J. Barrientos, C. Fredes, A. Valenzuela, "A review of convolutional neural network applied to fruit image processing", Applied Sciences, Vol. 10, No. 10, 2020, p. 3443.

[20] Y. Xu, J. Kou, Q. Zhang, S. Tan, L. Zhu, Z. Geng, X. Yang, "Visual Detection of Water Content Range of Sea buckthorn Fruit Based on Transfer Deep Learning", Foods, Vol. 12, No. 3, 2023, p. 550.

[21] J. L. Joseph, V. A. Kumar, S. P. Mathew, "Fruit classification using deep learning", Innovations in Electrical and Electronic Engineering: Proceedings of ICEEE, IEEE, 2021, pp. 807-817.

[22] A. Pande, M. Munot, R. Sreeemathy, R. V. Bakare, "An efficient approach to fruit classification and grading using deep convolutional neural network", Proceedings of the IEEE 5th International Conference for Convergence in Technology, Bombay, India, 29-31 March 2019, pp. 1-7.

[23] H. Yang et al. "A novel method for peanut variety identification and classification by Improved VGG16", Scientific Reports, Vol. 11, No. 2, 2021, p. 15756.

[24] Y. D. Zhang, Z. Dong, X. Chen, W. Jia, S. Du, K. Muhammad, S. H. Wang, "Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation", Multimedia Tools and Applications, Vol. 78, 2020, pp. 3613-3632.

[25] R. Dandavate, V. Patodkar, "CNN and data augmentation-based fruit classification model", Proceedings of the Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), Palladam, India, 7-9 October 2020, pp. 784-787.

Precipitation forecast using RNN variants by analyzing Optimizers and Hyperparameters for Time-series based Climatological Data

Original Scientific Paper

J. Subha
Centre for Information Technology and Engineering,
Manonmaniam Sundaranar University, Tirunelveli – 12, Tamil Nadu, India
subha.arhip@gmail.com

S. Saudia
Centre for Information Technology and Engineering,
Manonmaniam Sundaranar University, Tirunelveli – 12, Tamil Nadu, India
saudiasubash@msuniv.ac.in

Abstract *– Flood is a significant problem in many regions of the world for the catastrophic damage it causes to both property and human lives; excessive precipitation being the major cause. The AI technologies, Deep Learning Neural Networks and Machine Learning algorithms attempt realistic solutions to numerous disaster management challenges. This paper works on RNN-based rainfall/ precipitation forecasting models by investigating the performances of various Recurrent Neural Network (RNN) architectures, Bidirectional RNN (BRNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU) and ensemble models such as BRNN-GRU, BRNN-LSTM, LSTM-GRU, BRNN-LSTM-GRU using NASAPOWER datasets of Andhra Pradesh (AP) and Tamil Nadu (TN) in India. The different stages in the workflow of the methodology are Data collection, Data pre-processing, Data splitting, Defining hyperparameters, Model building and Performance evaluation. Experiments for identifying improved optimizers and hyperparameters for the time-series climatological data are investigated for accurate precipitation forecast. The metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Square Error (RMSE) and Root Mean Squared Logarithmic Error (RMSLE) values are used to compare the precipitation predictions of different models. The RNN variants and ensemble models, BRNN, LSTM, GRU, BRNN-GRU, BRNN-LSTM, LSTM-GRU, BRNN-LSTM-GRU produce predictions with RMSLE values of 2.448, 0.555, 0.255, 1.305, 1.383, 0.364, 1.740 for AP and 1.735, 0.663, 0.152, 0.889, 1.118, 0.379, 1.328 for TN respectively. The best performing RNN model, GRU when ensembled with the existing statistical model SARIMA produces an RMSLE value of 0.754 and 1.677 respectively for AP and TN.*

Keywords*:* *deep Learning, optimizers, hyperparameters, RNN, BRNN, LSTM, GRU, SARIMA*

Received: November 1, 2023; Received in revised form: February 9, 2024; Accepted: February, 12 2024

1. INTRODUCTION

Around the world, floods regularly cause enormous losses; India also suffers the most serious damages [1, 2]. The prime cause being excessive precipitation [3, 4] that occurs suddenly resulting in hazardous flood conditions and difficulties for the people. Out of all natural disasters, floods in India account for 56% of all fatalities [4]. According to Central Water Commission (CWC) data for India, between 1953 and 2020 there was an average of 1676 flood-related fatalities every year. Table 1 depicts the Flood-affected areas and flood damages in India during the period from 1953 to 2020 [5].

In the statement on the climate of India released by the India Meteorological Department (IMD) in the year 2022 [6] concerning the data from 1971 to 2020, the majority of India had a high long-period average (LPA) precipitation of 108%. During the years, South Peninsular India received seasonal monsoon precipitation equal to 122% LPA; Central India and Northwest India received seasonal precipitation equal to 119% and 101% LPAs respectively; and East & Northeast India received seasonal monsoon precipitation equal to 82% LPA [7]. The heavy rainfall days showed significantly increasing flood trends over peninsular India [8]. The impact of excessive rainfall is the cause of flood frequency in various parts of the world especially India, Indonesia and Spain [8-11]. Also, from [9] and [11], the amount of daily rainfall turned out to have a stronger correlation with floods. This thought motivates the research to make flood forecasts for safe living in extreme rainfall-receiving areas of India. Rainfall events are classified as moderate when it is between 2.5 and 64.5 mm/day, while events that fall beyond 64.5 mm/day are classified as heavy rainfall [8]-[10]. So, when the rainfall is over 64.5 mm/day, there are chances of flooding. Thus, this research which involved rainfall prediction is important to forecast flood risk based on predicted rainfall values greater than 64.5 mm/day.

Flooding is possible in heavy precipitation-receiving regions like South Peninsular India or South India. The union territories of the Andaman and Nicobar Islands, Lakshadweep and Puducherry are included in South India in addition to the Indian states of Andhra Pradesh, Karnataka, Kerala, Tamil Nadu and Telangana. South India encompasses 20% of the nation's population and 19.31% of its total area (635,780 km2 or 245,480 sq. miles) [12]. It is well recognized that anticipating flood disasters requires an awareness and analysis of variations in precipitation [13]. The purpose of this research is to analyze time-series climatological data and execute RNN-based precipitation forecast models to prevent flooding in two of the states in South Peninsular India: Andhra Pradesh (AP) and Tamil Nadu (TN). The daily precipitation values of TN and AP for the last 20 years from 2002 to 2022 as identified from the NASAPOWER dataset [14], range from 0 to 178.98 mm/day.

Table 1. Flood damages in India from1953 to 2020

Description of Flood Damages	Measure Unit	Total damage	Average damage
Area Affected	million hectares (m.ha.)	493	7.24
Population Affected	million	2199	32.34
Crops Area Affected	m. ha.	276	4.06
Values of Crops Affected	Rs. Crore	131462	1933.27
Number of Houses Damaged	Nos.	82525198	1213606
Value of Damaged Houses	Rs. Crore	57018	838.50
Number of Cattle Lost	Nos.	6182943	90926.00
Number of Human Lives Lost	Nos.	113943	1676.00
Value of Damaged Public Utilities	Rs. Crore	234149	3443.37
Value of Total Damages (Inc. Houses, Crops, public utilities)	Rs. Crore	437150	6428.67

Presently, many technologies, including Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) provide solutions for different domains of disaster management. ML algorithms, such as Regression [15], Support Vector Machine (SVM) [7], Decision Trees (DT) [16, 17], Naive Bayes [18] and K-Nearest Neighbors (KNN) [19] have been effectively used to construct precipitation prediction or classification models in a variety of domains. DL algorithms, such as Artificial Neural Networks (ANN) [20], Recurrent Neural Networks (RNN) [21, 22], Convolutional Neural Networks (CNN) [23] and Generative Adversarial Networks (GAN) [24] play an essential role in processing and analyzing massive amounts of precipitation data to deliver meaningful information. Venkatesh, et al. (2021) [25] constructed a precipitation prediction system using GAN with a CNN upon time-series annual precipitation data of 36 subdivisions in India from 1901 to 2015. This provided a better prediction for summer, winter, pre-monsoon, and post-monsoon precipitations with an accuracy of 99%. S Aswin, et al. (2018) [26] developed a model for predicting precipitation using DL architectures, LSTM and CNN for the Global Precipitation Climatology Project (GPCP) upon a monthly precipitation dataset that encompasses the time frame of July 1979 to January 2018. Also, it declared that LSTM and CNN produced RMSE of 2.55 and 2.44 respectively.

Haq, et al. [27] constructed an LSTM model based on El-Nino and Indian Ocean Dipole (IOD) precipitation data to predict precipitation for Sidoarjo, East Java and Indonesia. With a hidden layer, batch size and drop period of 100, 32 and 150 respectively, the model performed with a Mean Arctangent Absolute Percentage Error (MAAPE) value of 0.5810. Dechao et al. [28] created a DL-based forecast on the intensity of the regional precipitation in the next two hours with the use of 3 dimensional CNN (Conv3D), GRU algorithms and radar images. Ouma, et al. [29] presented precipitation and time-series trend analysis using LSTM and Wavelet Neural Networks (WNN). Moreover, predictions were made using hydrologic basin precipitation streamflow data and satellite-based meteorological data from 1980–2009. Both models performed well and predicted the precipitation with R2 values of 0.8610 and 0.7825 respectively. Samad et al. [30] built a model for rainfall prediction based on an Australian dataset for the regions of Albany, Walpole, and Witchcliffe. The LSTM network outperformed the ANN after comparison using different performance measures including MSE, RMSE and MAE. The LSTM model accurately predicted the precipitation with RMSE values of 5.343, 6.280, and 7.706 for the three regions respectively. Dada et al. [21] proposed four Neural Network models: Feed Forward Neural Network (FFNN), RNN, Elman Neural Network (ENN) and Cascade Forward Neural Network (CFNN) for predicting precipitation using India's precipitation data from the Kaggle repository. Additionally, it is clear from the statistical findings that the ENN model outperformed the other three models. ENN was discovered to have the best performance with the lowest RMSE, MSE, and MAE values of 6.360, 40.45, and 0.54 respectively. Saha et al. [31] used an ensemble regression tree model utilizing data from 1948 to 2015 for estimating monsoon precipitation over homogeneous regions of India. Forecast errors for the monsoons in central, northeast, north-west and south-peninsular India are 4.1%, 5.1%, 5.5% and 6.4% respectively.

From the baseline references [1-5], it is found necessary to build an effective flood warning system to prevent recurring harm sustained by people. Also, precipitation is the most significant reason for generating floods. Time-series data-based precipitation forecast can aid in the decision-making processes associated with flood and disaster to manage, flood, control floods and plan safety preparations [31, 32].

Also, based on a survey of related work [21, 25-31], it has been determined that articles for predicting precipitation use DL neural network models such as LSTM, ANN, GRU, FFNN, RNN, ENN, and CFNN. Also, measures such as RMSE and MAE are used to assess the expected precipitation. Thus, to assist in the decision-making procedures for preventing floods which demand extremely precise predicted precipitation, the study decides to carry out experiments and a thorough analysis by experimenting with a variety of optimizers and hyperparameters to forecast precipitation using RNN variants on time-series precipitation data. This study focuses on creating RNN variant models using BRNN, LSTM, GRU and ensemble models such as BRNN-GRU, BRNN-LSTM, LSTM-GRU and BRNN-LSTM-GRU to improve the effectiveness of RNN-based precipitation forecast. Finally, the paper also attempts an ensemble model for predicting precipitation using the lowest-error models of RNN variants with the statistical model, SARIMA. The most effective optimizer and hyperparameters as identified from the experiments are selected to predict precipitation using the RNN variant model. Also, the results of the proposed RNN variants and ensemble models are compared in terms of error metrics, with those from publications by S. Aswin et al. [26], Haq et al. [27], Ouma et al. [29], Samad A, et al. [30], Dada et al. [21], and Saha et al. [31]. The comparative analysis demonstrates that the proposed model outperforms all other models and techniques under comparison. The rest of this paper is organized in three more sections: Section 2 describes the workflow of the proposed methodology; Section 3 presents the results of the experiments and Section 4 summarizes the findings of the proposed work.

2. PROPOSED SYSTEM

The proposed methodology for designing the models for daily precipitation forecasting for the southern states of AP and TN is shown in Fig. 1. The proposed system predicts precipitation for AP and TN using RNN variants such as BRNN, LSTM, GRU, and ensemble models such as BRNN-GRU, BRNN-LSTM, LSTM-GRU and BRNN-LSTM-GRU with suitable hyperparameters and optimizers. The different stages in the workflow of the methodology are Data collection, Data pre-processing, Data splitting, Defining hyperparameters, Model building and Performance evaluation. Time-series-based precipitation data from 2002 to 2022 is downloaded from the NASAPOWER website- https://power.larc.nasa.gov/ [14].

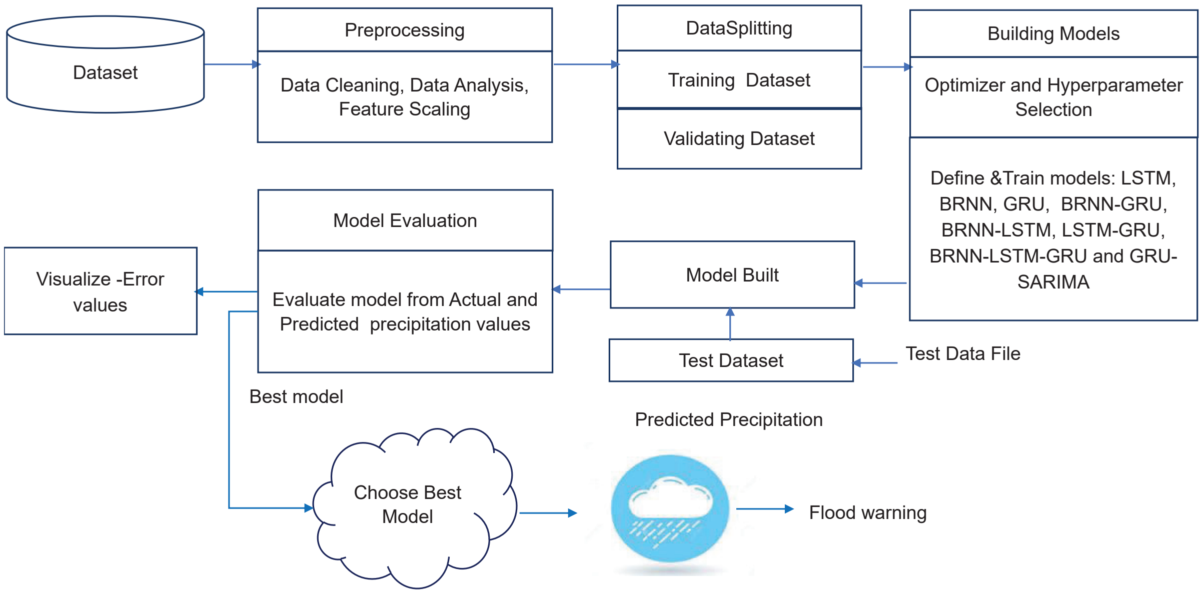

Dataset
Preprocessing
Data Cleaning, Data Analysis, Feature Scaling
DataSplitting
Training Dataset
Validating Dataset
Building Models
Optimizer and Hyperparameter Selection
Define &Train models: LSTM, BRNN, GRU, BRNN-GRU, BRNN-LSTM, LSTM-GRU, BRNN-LSTM-GRU and GRU-SARIMA
Model Evaluation
Evaluate model from Actual and Predicted precipitation values
Visualize -Error values
Model Built
Test Dataset
Test Data File
Best model
Predicted Precipitation
Choose Best Model
Flood warning

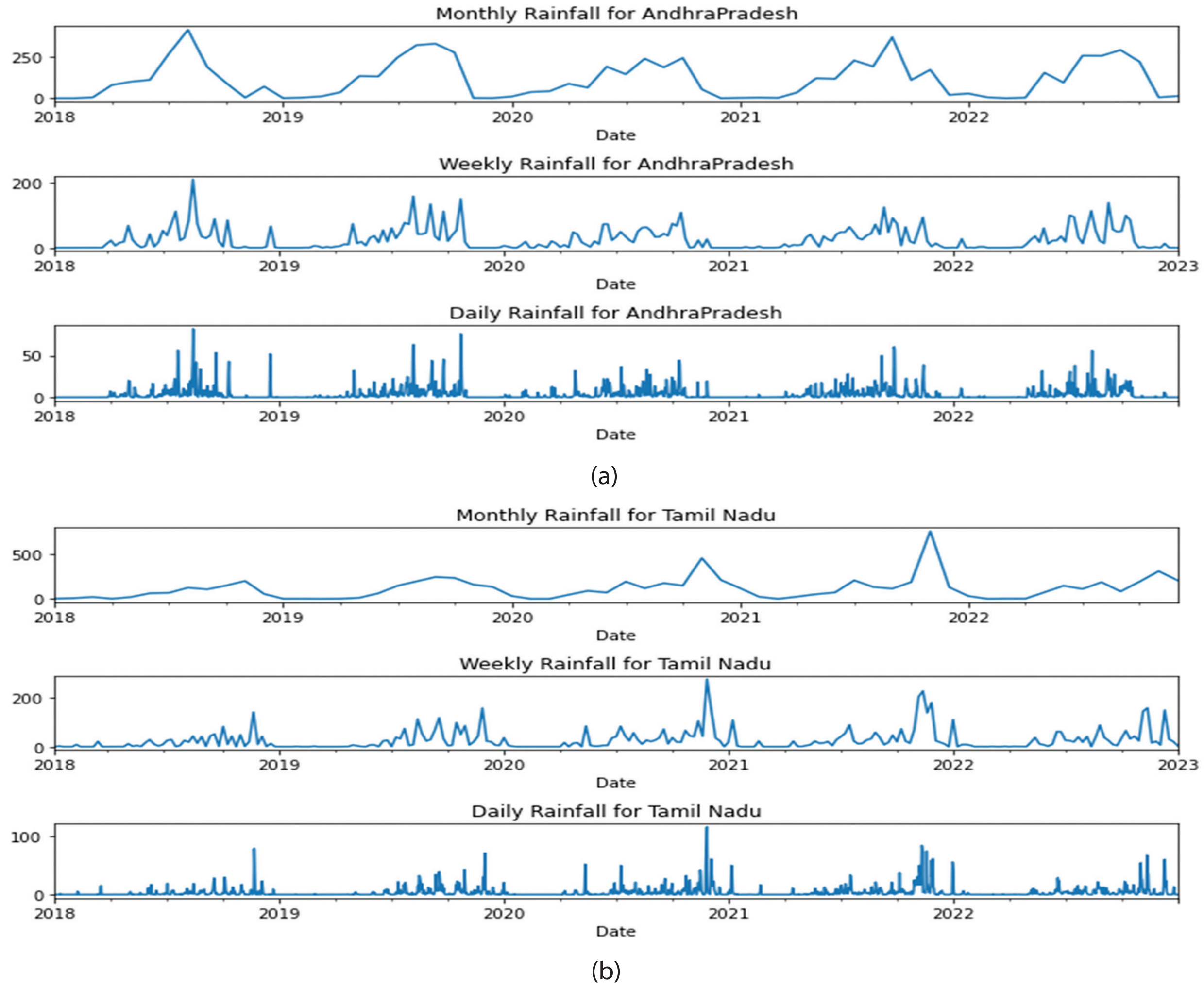
Fig. 1. Methodology of the proposed RNN based precipitation forecast system

The best hyperparameters and optimizers for the RNN variants, BRNN, LSTM and GRU are identified and used to build the precipitation forecast models. Several ensemble models like BRNN-GRU, BRNN-LSTM, LSTM-GRU and BRNN-LSTM-GRU are also experimented with for predicting precipitation. The performances of different RNN and the ensemble models are evaluated using MSE, RMSE, MAE and RMSLE. Finally, a GRU-SARIMA is created with the lowest error-producing GRU and the existing statistical model SARIMA for predicting precipitation.

2.1. DATASET

The time-series data is obtained from the website https://power.larc.nasa.gov/ [14] for training the DL and ensemble algorithms. This website provides climatological data obtained from satellite observations of agricultural and renewable energy usage. The time-series climatological dataset is collected for the southern Indian states of AP and TN in Comma-Separated Value [CSV] file format for the years from 2002 to 2022. This time-series dataset had 7670 records. In this study, only the precipitation data and the corresponding date of year are taken from the dataset. The date of year feature is created using the to_datetime () function in the pandas package of Python. This forms a univariate time series of precipitation data covering 20 years from 2002 to 2022. Utilizing the Python packages pandas and matplotlib, the dataset is visualized as in Fig. 2 which shows the daily, weekly and monthly precipitation totals of AP and TN states for the years 2002–2022. Fig. 2 depicts the fluctuation in precipitation and the peak points corresponding to the highest and lowest precipitation.

Every year from June to December, a time series pattern in the form of increasing trends in precipitation is seen in the TN and AP statistics.



(a)

(b)

Fig. 2. Line Plots showing increasing trends in the daily, weekly, and monthly precipitation statistics of a) Andhra Pradesh b) Tamil Nadu from 2018 to 2022

Fig. 2 illustrates the seasonal pattern in the precipitation data for the AP and TN states. Thus, the use of RNN variant models, which are univariate time series forecasting models is suggested for flood prediction since RNN variants capture time dependency in the data and so are better at prediction than other models. Additionally, time-series data is particularly well-suited for RNN [33, 34]. The likelihood of catastrophic flooding increases with heavy precipitation and so precipitation forecasting is necessary to prevent severe casualties.

2.2. DATA PREPROCESSING

Pre-processing is done on the data to remove any null, empty or outlier data or to replace them with the right data before training RNN and ensemble algorithms. Extreme data values that lie outside the observation range are outliers. To eliminate data inconsistencies, outliers and missing numbers are corrected/ filled in and the dataset is formatted for training [35-37]. No similar pre-processing is required because the climatological data obtained NASAPOWER website [14] for AP and TN is free of missing values and outliers.

The precipitation values in the dataset range from 0 to 178.98. To improve model efficiency, min-max scaling as defined in equation (1) is applied to the attribute, precipitation.

$$sc_i = \frac{\min(y_i)}{\max(y_i) - \min(y_i)} \quad (1)$$

MinMaxScalar function in the sklearn package of Python produces the scaled data, SC_i in the range, [0,1] for every i^{th} precipitation observation, y_i. The scaled dataset is divided into training and validation datasets in an 80:20 ratio [36, 37] with the training dataset being used to train the aforementioned RNN variants and ensemble techniques. The performance of the models is evaluated using the validation dataset [35]. The preprocessed univariate time-series dataset is then subjected to the training phase of the RNN algorithms in the Model Building stage.

2.3. MODEL BUILDING

This study proposes to design and analyze precipitation forecast models using time series precipitation data of TN and AP states with variant RNN architectures: the BRNN, LSTM and GRU and ensemble techniques, BRNN-GRU, BRNN-LSTM, LSTM-GRU, BRNN-LSTM-GRU and GRU-SARIMA.

2.3.1. RNN ARCHITECTURES

Recurrent Neural Network is a kind of DL Neural Network; numerous applications with time series data have successfully used the RNN [38]. An RNN uses multiple layers of neurons to construct a model based on training data to forecast unknown or future data. Three basic RNN architectures exist: BRNN [36], LSTM and GRU [39, 40]. A review of these architectures is made in the following sub-sections.

2.3.2. BRNN

The Bidirectional Recurrent Neural Network (BRNN) is a variant of RNN architecture. While unidirectional RNNs can only utilize previous inputs to predict precipitation, BRNN works to increase the forecast accuracy by focusing on the previous and subsequent situations as shown in Fig. 3. It has two RNNs that are oriented in opposite directions and linked to the same output layer. The BRNN receives not only the hidden layer output of the previous moment as an input but also the hidden layer output of the following moment [23].

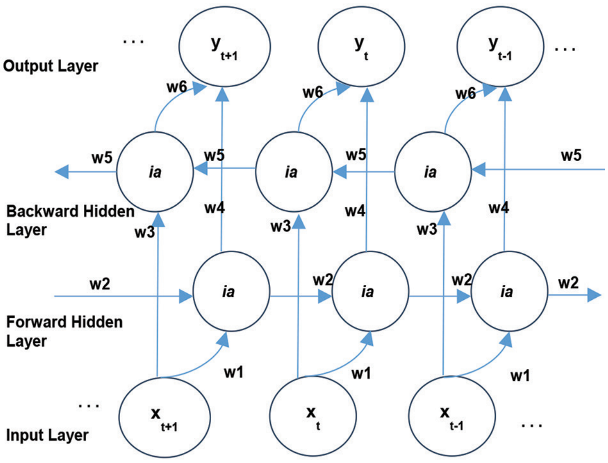


Fig. 3. Structure of BRNN

The values in the hidden and output layer neurons are determined in the forward pass of training based on the equations from (2) to (4).

$$h_t(F) = ia(x_t * w_1(F) + h_{t-1}(F) * w_2\,(F) + b_h(F) \quad (2)$$

$$h_t(B) = ia(x_t * w_3(B) + h_{t+1}(B) * w_5\,(B) + b_h(B) \quad (3)$$

$$y_t = h_t(F) * w_4 + h_t(B) * w_6 + b_y \quad (4)$$

Here, x_t is the input with values x_1, x_2, x_3… at time slots, t=1,2,3,…, h_t is the hidden state memory values, h_1, h_2, h_3, ….. at time slots, t=1,2,3,…, ia is the hidden layer activation function, w_1, w_2 are the weights associated with forward hidden layer and w_3, w_4 are the weights associated with the backward hidden layer. $b_h(F)$ and $b_h(B)$ are biases to the forward and backward layers respectively. $h_t(F)$ is the output from the forward hidden layer of $h_t(B)$ is the output from the backward hidden layer. y_t is the precipitation output produced for the instant, 't'. The hyperparameters and optimizers used in the forward and backward passes of the training phase of the BRNN obtained after fine-tuning are mentioned in Table 6 and a discussion on the precipitation forecast is provided in Subsection 3.

2.3.3. LSTM

Sepp Hochreiter and Juergen Schmidhuber introduced the Long Short-Term Memory (LSTM) as an alternative architecture of RNN in 1997 [39, 41]. A typical LSTM structure is made up of a cell unit and three major gates: an input gate, a forgetting gate and an output gate as shown in Fig. 4 (a). The cell unit is a memory unit that can store information for a long period. The memory unit's writing, reading and saving are controlled in order by the input gate, output gate, and forgetting gate. When the forgetting gate outputs 1, the cell unit writes and saves the information; when the forgetting gate outputs 0, the cell unit deletes the saved information; when the input gate outputs 1, the rest of the LSTM reads the cell unit information; and when the input gate outputs 0, the rest of the LSTM writes the contents to the cell unit.

The values of the gates in the hidden layer and output values are determined in the forward pass of training based on the equations from (5) to (11).

$$f_t = \sigma\,(x_t * u_f + h_{tlstm-1} * w_f) \quad (5)$$

$$\bar{C}_t = ia(x_t * u_c + h_{tlstm-1} * w_c) \quad (6)$$

$$i_t = \sigma\,(x_t * u_i + h_{tlstm-1} * w_i) \quad (7)$$

$$o_t = \sigma\,(x_t * u_o + h_{tlstm-1} * w_o) \quad (8)$$

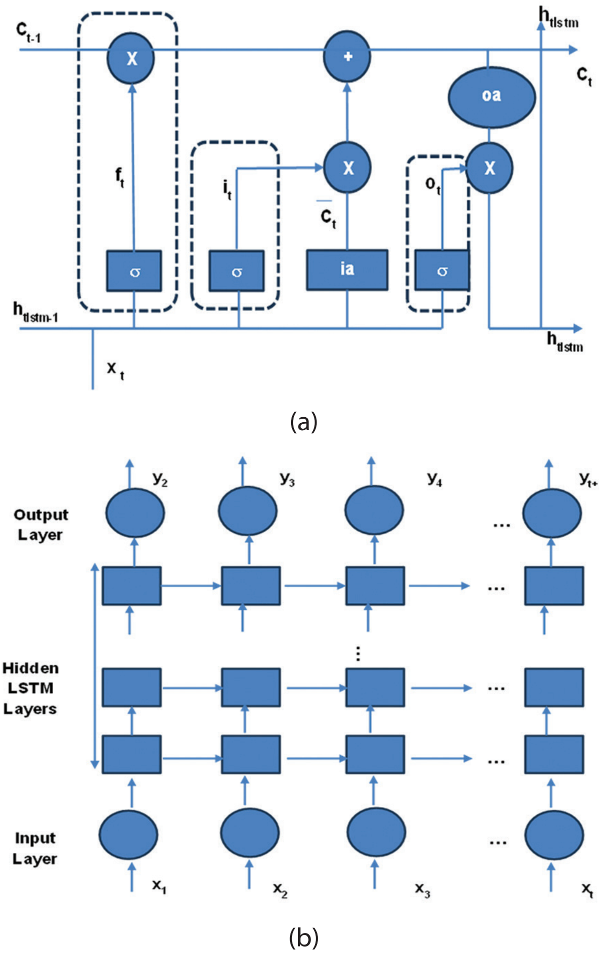
$$C_t = (f_t * C_{t-1} + i_t * \bar{C}_t) \quad (9)$$

$$h_{tlstm} = o_t * ia(C_t) \quad (10)$$

$$y_t = oa\,(w_t * h_{tlstm} + b\,) \quad (11)$$

If x_t is the input with values x_1, x_2, x_3… at time slots, t=1,2,3,…, the forget gate, f_t, input gate, it and output gate, ot values are determined from the equations (5), (7) and (8) respectively. The long-term memory state, C_t and the hidden state memory values, h_{1lstm}, h_{2lstm}, h_{3lstm},…..at time slots, t=1,2,3,…, are determined as in equations (9) and (10) respectively. Here w_f, w_c, w_i and wo are weights associated with the link carrying hidden state information, h_{1lstm} to the forget gate, long-term memory state, input gate and output gate respectively. Also, u_f, u_c, u_i and u_o are the weights associated with links carrying input, x_t to the forget gate, long-term memory state, input gate and output gate respectively. b is the bias associated with output neuron y_t corresponding to the timeslot t. The precipitation forecast of LSTM at different time slots, 't' is determined as in equation (11). σ ia and oa are the *sigmoid*, *tanh* and linear activation functions.

Fig. 4 (b) shows a deep LSTM architecture where the h_{tlstm} value produced by a hidden neuron at a particular level is the input to the next hidden neuron of that level. The output neuron of that level produces the output precipitation value, y_t as a linear function of htlstm value from its immediately previous hidden neuron as in equation (11) using the weight, 'w_t'. The model is trained using the data from NASAPOWER training datasets of AP and TN with the hyperparameters listed in Table 6 for precipitation forecast. A discussion on the test results is provided in Subsection 3.



(a)

(b)

Fig. 4. Structure of a) basic LSTM cell b) deep LSTM

2.3.4. GRU

Another RNN variant is the Gated Recurrent Unit (GRU). It is an upgraded and enhanced version of LSTM which debuted in 2014 [15, 42]. Compared to the three gates of an LSTM, it requires fewer parameters because of the reset gate, rt and update gate, zt. The decision of which information should be transmitted to the output is made using zt and rt, which are vectors as shown in Fig. 5. This study uses the GRU architecture also to predict daily precipitation.

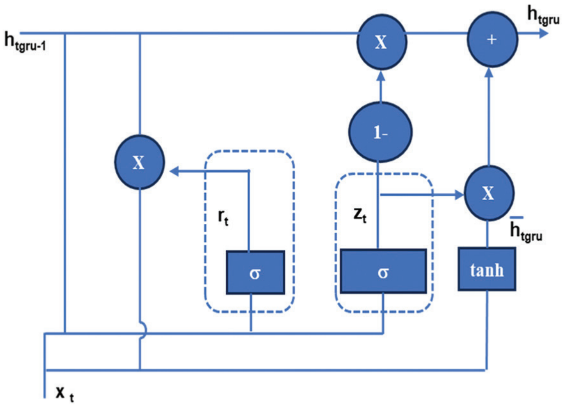


Fig. 5. Structure of basic GRU cell

The values of the gates in the hidden layer and output values are determined in the forward pass of training based on the equations from (12) to (16).

$$z_t = \sigma\,(w_z * \big(h_{tgru-1}, x_t\big)) \qquad (12)$$

$$r_t = \sigma\,(w_r * \big(h_{tgru-1}, x_t\big)) \qquad (13)$$

$$\bar{h}_{tgru} = ia\,(w * (r_t * \big(h_{tgru-1}, x_t\big)) \qquad (14)$$

$$h_{tgru} = (1{-}z_t) * \big(h_{tgru-1} + z_t * \bar{h}_{tgru}\big) \qquad (15)$$

$$y_t = oa\,(w_t * h_{tgru} + b\,) \qquad (16)$$

If x_t corresponds to the input values x_1, x_2, x_3… at time slots, t=1,2,3,…. Reset gate, r_t and update gate, z_t values are determined from the equations (12) and (13) respectively. The hidden state memory values, h_{1gru}, h_{2gru}, h_{rgru}, ..., at time slots, t=1,2,3,…, are determined as in equations (15). The precipitation forecast of GRU at different time slots is in equation (16). The weights w_z, w_r and w are associated with update gate, reset gate and previous hidden state respectively. b is the bias associated with output neuron, y_t of timeslot t. σ ia and oa are the sigmoid, tanh and linear activation functions.

Deep GRU architecture is similar to deep LSTM as shown in Fig 4 (b), where the h_{tgru} value produced by a hidden neuron at a particular layer is the input to the next hidden neuron of that level. The output neuron of that level produces the output precipitation value, y_t as a linear function of h_{tgru} value from its immediately previous hidden neuron as in equation (16). In eqn. (16), wt is the weight associated with the output neuron, y_t.

The model is trained using the training datasets from AP and TN with the hyperparameters listed in Table 6 which are obtained by analyzing various hyperparameters and optimizers. Also, a discussion of the predicted precipitation is provided in Subsection 3.

2.3.5. ENSEMBLE ARCHITECTURE

This section discusses the ensemble models that incorporate the RNN variant models: BRNN, LSTM and GRU. Averaging, Bagging, and Stacking are the three ensemble learning techniques [43]. Ensemble learning techniques are often trained using many models on the same dataset utilizing each learned model to generate a forecast and combining the results of the individual model to produce the final result or forecast [43]. In this work, an average ensemble of selected RNN models is determined based on the results of the models to produce accurate precipitation forecasts as shown in Fig. 6.

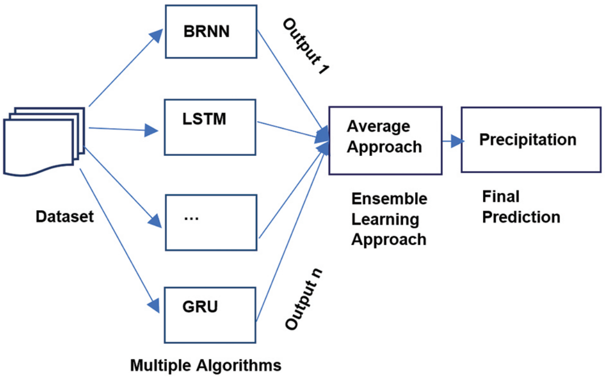


Fig. 6. General Workflow for Ensemble Techniques

The different ensemble models used in the analysis are BRNN-GRU, BRNN-LSTM, LSTM-GRU and BRNN-LSTM-GRU as shown in Fig. 7. The optimized hyperparameters used by LSTM, GRU and BRNN as mentioned in Table 6 are used by the base models of the ensemble techniques also. The performance of the models BRNN, LSTM, GRU, BRNN-GRU, BRNN- LSTM, LSTM-GRU and BRNN-LSTM-GRU are provided in Table 7 and 8.

The RNN variant with the lowest error, GRU is also used to create a hybrid ensemble model with statistical SARIMA for precipitation forecasting as shown in Fig. 8. SARIMA is a Seasonal AutoRegressive Integrated Moving Average model for time-series data-based forecasting that uses past values. SARIMA is an advanced version of AutoRegressive Integrated Moving Average (ARIMA) [7] for Time Series forecasting based on its past values, lags and lagged forecast error values along with the seasonal characteristics of the time-series data with seasonal patterns.

The SARIMA model has the parameters: the order of the 'Auto Regressive' (AR) phrase, the number of lags to be utilized as predictors and the order of the moving average along with the parameters of seasonal characteristics which includes: seasonal autoregressive order, seasonal difference order, seasonal moving average order and the periodicity of seasons [10]. The precipitation is found by averaging the forecasts of the GRU model and the SARIMA model. The ensemble model is trained using historical precipitation data of TN and AP from 2002 to 2022. By leveraging its learned patterns, it predicts the precipitation for the next 30 days, providing valuable insights for planning and decision-making against floods. Additionally, the performance of this hybrid ensemble model is assessed in terms of MAE, MSE, RMSE and RMSLE. In Section 3, the outcomes of these RNN variant models and ensemble models are examined in terms of evaluation metrics, MAE, MSE, RMSE, and RMSLE. The values are shown in Tables 7 and 8.

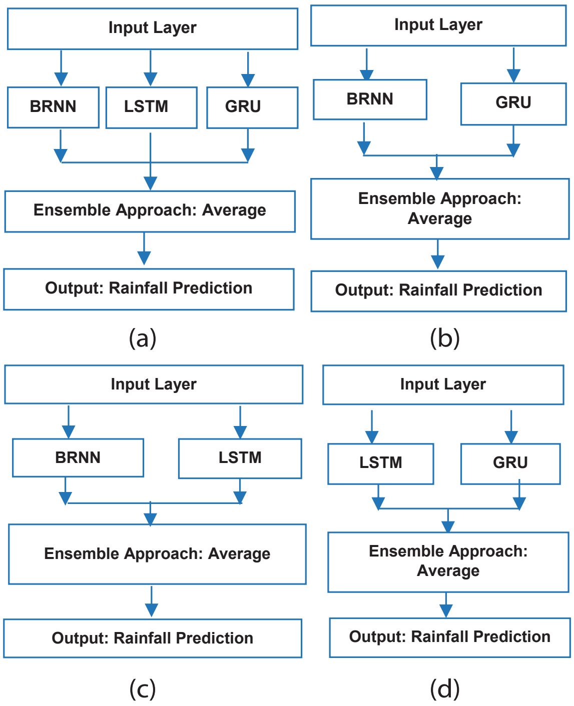

Input Layer
BRNN
LSTM
GRU
Ensemble Approach: Average
Output: Rainfall Prediction
(a)
Input Layer
BRNN
GRU
Ensemble Approach: Average
Output: Rainfall Prediction
(b)
Input Layer
BRNN
LSTM
Ensemble Approach: Average
Output: Rainfall Prediction
(c)
Input Layer
LSTM
GRU
Ensemble Approach: Average
Output: Rainfall Prediction
(d)

Fig. 7. Architecture of RNN-based Ensemble models a) BRNN-LSTM-GRU b) BRNN-GRU c) BRNN-LSTM d) LSTM-GRU

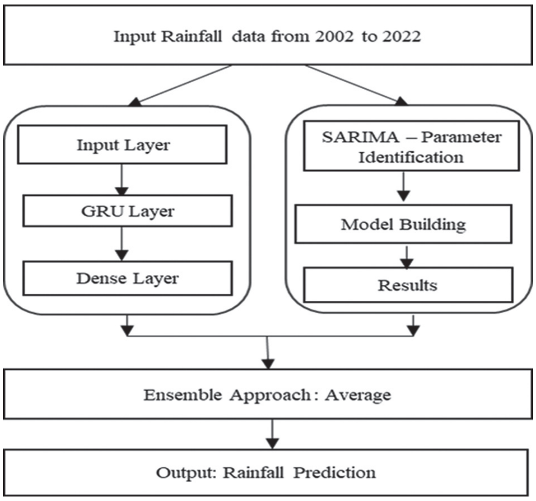
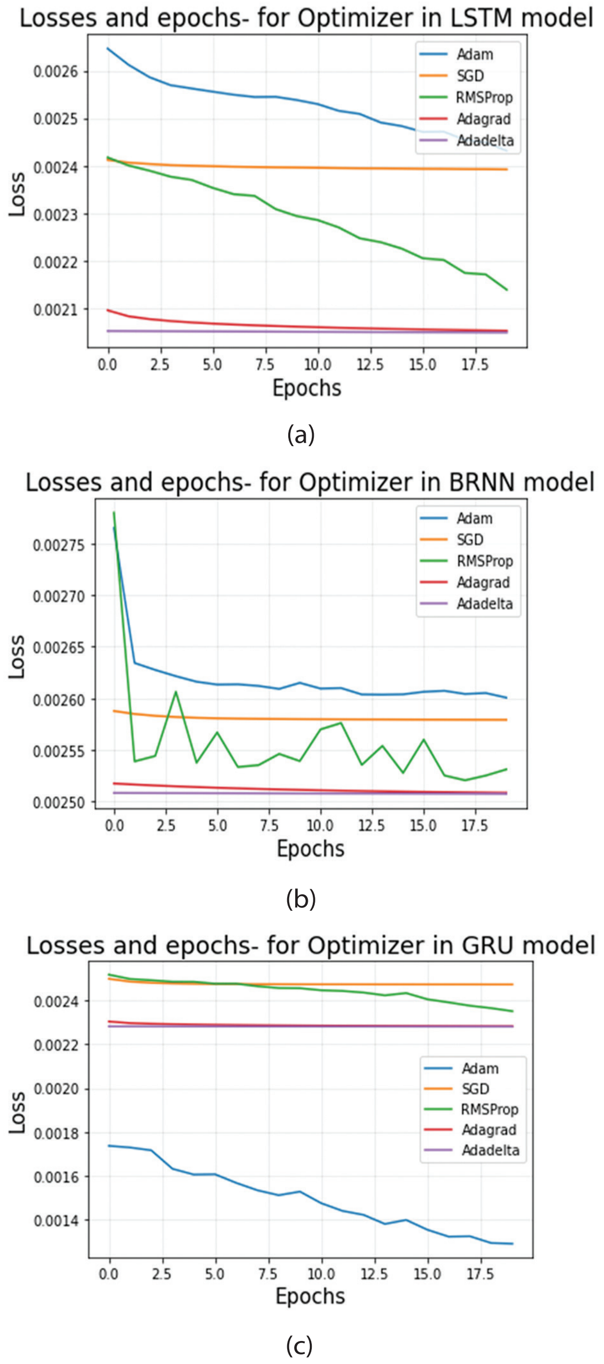

Input Rainfall data from 2002 to 2022
Input Layer
GRU Layer
Dense Layer
SARIMA – Parameter Identification
Model Building
Results
Ensemble Approach : Average
Output: Rainfall Prediction

Fig. 8. Block diagram showing GRU-SARIMA ensemble model



(a)

(b)

(c)

Fig. 9. Optimizer based losses in RNN variants a) LSTM b) BRNN and c) GRU

These models are developed using the DL and essential packages in Python namely TensorFlow, NumPy, Pandas, Matplotlib, Sklearn and Keras. Appropriate hyperparameters are essential to achieve the best results from the RNN variants and the ensemble models [43-45]. Hyperparameters are variables that control the neural network architecture and performance during training. Some of the hyperparameters are the number of layers, activation functions, loss function, batch size and optimizer. Optimizers and hyperparameters have been studied in this work to improve the accuracy of the precipitation forecasts from the RNN and ensemble models. The working steps for selecting hyperparameters and optimizers are explained in the following paragraph.

The RNNs variants used in this study are compared against various optimization models including Adaptive Moment Estimation (Adam) [46], Stochastic Gradient Descent (SGD) [46], Adaptive Gradient Descent (AdaGrad) [46], Extension of AdaGrad (Adadelta) [40, 46] and Root Mean Square Propagation (RMSProp) [40, 46] to determine which optimization model offers the best learning and prediction. For the same set of hyperparameters as listed in Table 2, different optimizers are tried for all the RNN variants. The performance is assessed in terms of RMSLE by making forecasts on the validating dataset as shown in Table 3. The losses are illustrated in Fig. 9 a) LSTM, b) BRNN, and c) GRU respectively for different epochs of the training.

Table 2. List of Hyperparameters for Selecting Optimizer

Hyperparameters	Value of Hyperparameters
Batch Size	32
No of epochs	20
No of Hidden Layers	1
Hidden Units	100
Activation Function	Tanh
Output-Units	30
Output-Layer-Activation-Function	Linear
Loss Function	MAE

Of all the optimizers, the Adam optimizer generated smaller and steadily decreasing losses while using the training dataset. Unlike Adam, the other optimizers delivered constant losses and substantial forecast errors. Even while the RMSProp offered continuously decreasing losses, the losses in the BRNN model fluctuate. As a result, in this study, the Adam optimizer is employed to forecast precipitation from all the RNN variants and ensemble models. Also, to select the best set of hyperparameters for RNN variants such as BRNN, LSTM, and GRU, a list of experiments with different hyperparameter combinations as mentioned in Table 4 are performed on all RNN variants.

Table 3. Evaluation of Optimizers based on RMSLE

Model	Optimizers				
	Adam	SGD	RMSProp	AdaGrad	Adadelta
LSTM	0.0017	0.0017	0.0020	0.0019	0.0019
GRU	0.0018	0.0018	0.0018	0.0018	0.0018
BRNN	0.0018	0.0018	0.0018	0.0018	0.0018

The RNN variants generated results as shown in Table 5 for different trials. The RMLSE values for BRNN, LSTM, and GRU in the trials from 1 to 4 are 0.26, 1.42, 0.26; 0.58, 1.39, 0.30; 0.57, 2.01, 0.48 and 0.97,1.10, 0.95 respectively. All trials from 1 through 4 are compared to select the best hyperparameters. The trials 1 and 2 are compared to select the best batch; trials 3 and 4 are compared to select the best epoch and the trials 1 and

4 are compared to select the best number of hidden layers. The performances of the different RNN variants in the above experiments are compared as shown in Fig.10 a) to d) of Section 3 in terms of MAE, MSE, RMSE and RMSLE. The most effective optimizer and hyperparameters as identified from the experiments and listed in Table 6 are selected for further training and testing to predict precipitation.

Table 4. List of Hyperparameters and Optimizer values for selecting best hyperparameters

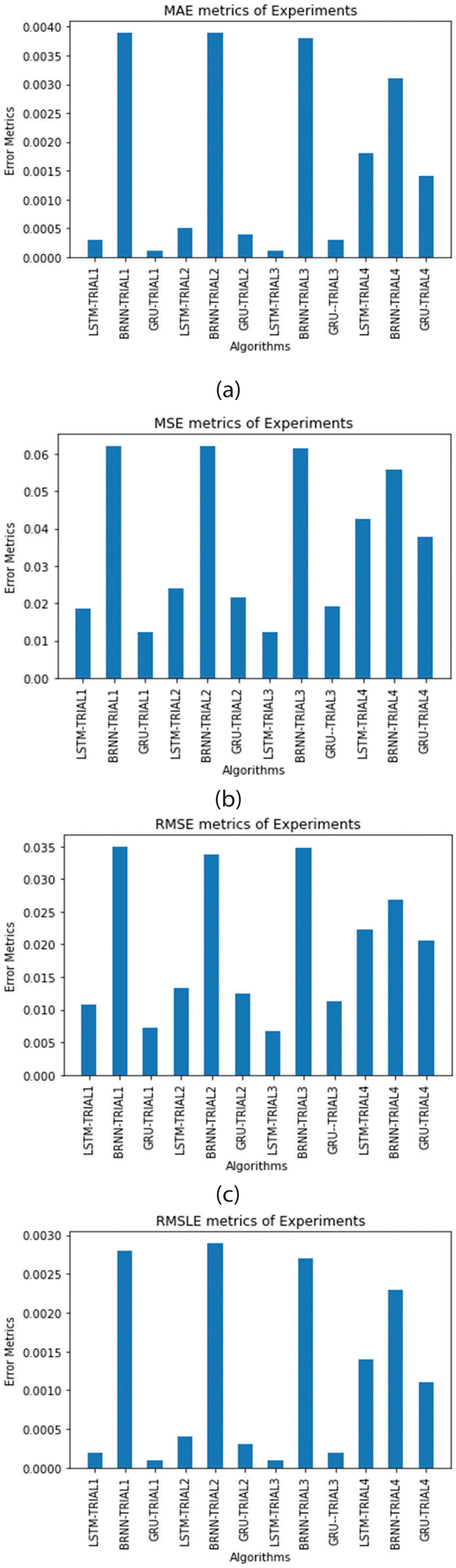
Hyperparameters	Trial 1	Trial 2	Trial 3	Trial 4
Number of Hidden Layers	3	3	1	1
Hidden units	128,256, 128	128, 256,128	128	128
Number of epochs	100	100	300	100
Batch size	32	64	32	32
Activation function	Tanh	Tanh	Tanh	Tanh
Optimizer	Adam	Adam	Adam	Adam
Loss function	MSE	MSE	MSE	MSE
Output-Units	30	30	30	30
Output-Layer-Activation-Function	Linear	Linear	Linear	Linear

Table 5. Comparison on the Performance of RNN variants in different trials

Model	Experiment	MAE	MSE	RMSE	RMSLE
BRNN	Trial1	0.0039	0.0621	0.0350	0.0028
LSTM		0.0003	0.0186	0.0107	0.0002
GRU		0.0001	0.0123	0.0072	0.0001
BRNN	Trial2	0.0039	0.0622	0.0337	0.0029
LSTM		0.0005	0.0241	0.0134	0.0004
GRU		0.0004	0.0216	0.0124	0.0003
BRNN	Trial 3	0.0038	0.0615	0.0348	0.0027
LSTM		0.0001	0.0122	0.0068	0.0001
GRU		0.0003	0.0193	0.0113	0.0002
BRNN	Trial 4	0.0031	0.0559	0.0269	0.0023
LSTM		0.0018	0.0426	0.0223	0.0014
GRU		0.0014	0.0377	0.0206	0.0011

Table 6. List of Hyperparameter values and optimizer selected for building BRNN, LSTM and GRU models

Hyperparameters	Value
Number of Hidden Layers	3 (128,256,128 units)
Number of epochs	300
Batch size	32
Activation function	Tanh
Optimizer	Adam
Loss function	MSE
Output-Units	30
Output-Layer-Activation-Function	Linear



(a)

(b)

(c)

(d)

Fig. 10. Bar plot showing the performances of RNN variants for different trials in terms of a) MAE b) MSE c) RMSE d) RMSLE

3. EXPERIMENTAL RESULTS AND DISCUSSION

This study underscores the importance of precipitation forecast to mitigate the negative effects of flood damage. Time-series data are often used to make predictions or forecasts of future values based on historical observations. The time-series data is obtained from the website, https://power.larc.nasa.gov/ [14] to forecast precipitation. Fig. 2 illustrates time-series data for days, weeks and months for AP and TN. However, day-wise data is used to train the deep learning RNN algorithms and ensemble techniques to predict the precipitation for the next 30 days. The model is created from the past 60 days' precipitation data to predict the precipitation of the next 30 days.

Experiments are conducted on an Intel Core i7 processor running at 2.70 GHz with 16GB of RAM using numerical and DL libraries of Python. TensorFlow, NumPy, Pandas, Matplotlib, Keras, and Sklearn are some of the packages used from Python. RNN based precipitation forecasting models are evaluated using the performance assessment metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Square Error (RMSE), and Root Mean Squared Logarithmic Error (RMSLE) values [30, 47]. The Mean Absolute Error (MAE) is the average of the absolute error difference as defined in Equation (17).

$$MAE = \frac{\sum_{i=1}^{m}|y_{ai} - y_{pi}|}{m} \quad (17)$$

The square root of the Mean Square Error (MSE) is referred to as the RMSE and is defined by Equation (18).

$$\boldsymbol{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_{ai} - y_{pi})^2} \quad (18)$$

where

$$\boldsymbol{MSE} = \frac{1}{m}\sum_{i=1}^{m}(y_{ai} - y_{pi})^2 \quad (19)$$

An RMSE variation that computes the logarithmic difference between the predicted and actual values is RMLSE. It is defined in equation (20).

$$\boldsymbol{RMSLE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\log(y_{ai} + 1) - \log(y_{pi} + 1))^2} \quad (20)$$

In all the above equations, the predicted value and the actual value corresponding to the i^{th} observation are denoted as y_{pi} and y_{ai} respectively. m is the number of records in the test dataset.

3.1. RESULTS AND DISCUSSION

In this work, for precipitation forecast, time-series based climatological data from 2002 to 2022 are downloaded from the NASAPOWER website. Subsequently, the data preprocessing operations are carried out. Afterward, the data is split in the ratio, 80:20 for data training and validation. The most effective hyperparameters and optimizer as identified and mentioned in Table 6 are chosen in building the RNN and ensemble models: BRNN, LSTM, GRU, BRNN-GRU, BRNN-LSTM, LSTM-GRU and BRNN-LSTM-GRU.

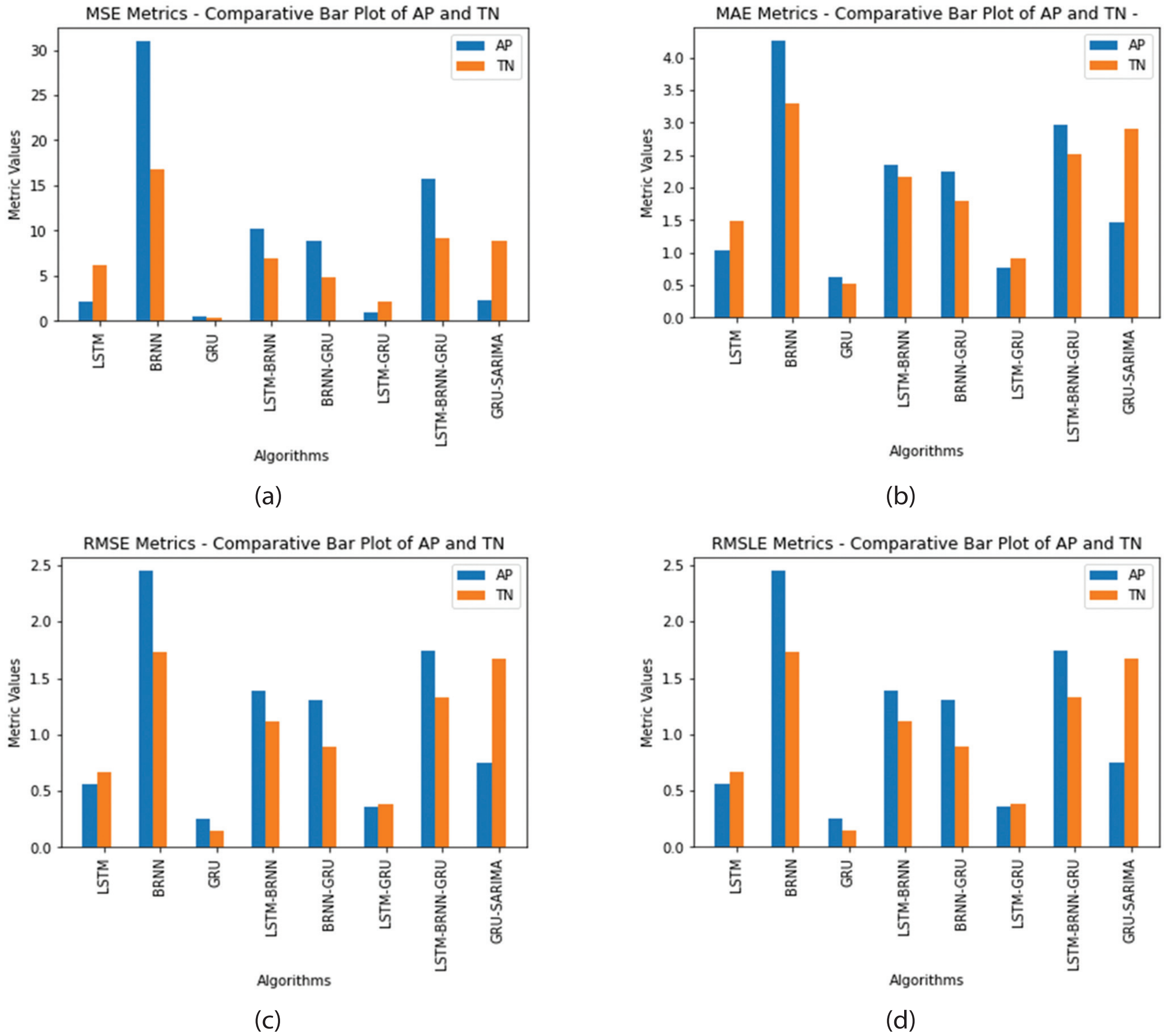
The test accuracy of predictions made by the models are evaluated using MSE, RMSE, MAE, and RMSLE. These evaluation metric values are presented in Table 8 and 9 and in Fig. 11 a) to d) for AP and TN respectively. Finally, using the RNN model, GRU which produces lowest error, a hybrid ensemble model is created with the statistical model, SARIMA to predict precipitation. The accuracy of this model is listed in Table 7 and 8 for AP and TN respectively. The GRU model produces improved performance than all the other models under comparison with a net RMSLE value of 0.255 and 0.152 for AP and TN datasets respectively. The ensemble model, LSTM-GRU proves to be the next best model from among all the ensemble models under comparison with a net RMSLE value of 0.364 and 0.379 for AP and TN datasets respectively. The best performing RNN model, GRU when ensembled with the existing statistical model SARIMA produces an RMSLE value of 0.754 and 1.677 respectively for AP and TN. The results of the proposed GRU model is also compared in terms of error metrics with those from publications as shown in Table 9. From the analysis, of all the models, the best performing GRU model appears reliable for flood defense due to heavy precipitation. The comparative analysis demonstrates that the GRU as identified from the proposed methodology outperforms all other models and techniques in terms of RMSE values.

Table 7. Comparison on the Performance of different RNN models and ensemble techniques on Test Dataset from AP

Models	MAE	MSE	RMSE	RMSLE
LSTM	1.041	2.113	1.453	0.555
BRNN	4.255	30.922	5.561	2.448
GRU	0.635	0.558	0.747	0.255
BRNN-LSTM	2.351	10.176	3.190	1.383
BRNN-GRU	2.240	8.843	2.974	1.305
LSTM-GRU	0.766	0.933	0.966	0.364
BRNN-LSTM-GRU	2.972	15.683	1.453	1.740
GRU-SARIMA	1.461	1.278	1.528	0.754

Table 8. Comparison on the Performance of different RNN models and ensemble techniques on Test Dataset from TN

Models	MAE	MSE	RMSE	RMSLE
LSTM	1.482	6.221	2.494	0.663
BRNN	3.296	16.772	2.494	1.735
GRU	0.519	0.352	0.593	0.152
BRNN-LSTM	2.160	6.945	0.593	1.118
BRNN-GRU	1.801	4.843	2.201	0.889
LSTM-GRU	0.910	2.130	1.460	0.379
BRNN-LSTM-GRU	2.525	9.209	2.494	1.328
GRU-SARIMA	2.905	8.905	2.484	1.677



(a)

(b)

(c)

(d)

Fig. 11. Bar plot showing the performances of different of RNN variants and ensemble techniques on Test Dataset from TN and AP in terms of a) MAE b) MSE c) RMSE d) RMSLE

Table 9. Comparison of results from different algorithms

Authors	Region & Dataset	Algorithms	Best RMSE Value
S Aswin, et al. (2018) [26]	Geographic location 10368 - [26]	LSTM, CNN	LSTM- 2.55, CNN- 2.44
Y.O. Ouma, et al. (2020) [29]	Kenya - [29]	LSTM, WNN	LSTM -17.22 WNN- 14.64
Samad, A. et al. (2020) [30]	Australia - [30]	LSTM	LSTM - 5.30
E Gbenga Dada et al. (2021) [21]	India - [21]	FFNN, RNN, ENN, CFNN	ENN -6.36
Proposed Methodology-GRU	India – [14]	RNN variant and Ensemble	GRU - 0.593

4. CONCLUSION

Predicting precipitation is an effective flood defense method that helps minimize the impact of high precipitation events and safeguard vulnerable areas. In this study, RNN-based precipitation forecast algorithms were trained and tested using the time-series-based climatological data of heavy rainfall receiving South Indian states of Tamil Nadu, Andhra Pradesh. The DL algorithms and the ensemble techniques: BRNN, LSTM, GRU, BRNN-GRU, BRNN-LSTM, LSTM-GRU, BRNN-LSTM-GRU and GRU-SARIMA were trained with the best set of hyperparameters and optimizers identified experimentally. In addition, the performances of these models were assessed in terms of MAE, MSE, RMSE, and RMSLE values. The GRU model proved to be the most effective model among all the models with net RMSLE values of 0.225 and 0.152 for AP and TN datasets respectively. The ensemble model, LSTM-GRU proved the next best model from among all the models under comparison with net RMSLE values of 0.364 and 0.379 for AP and TN datasets respectively. Also, hybrid ensemble model GRU-SARIMA proved to be the effective model, with net RMSLE values of 0.754 and 1.677 for AP and TN datasets respectively. Thus, the analysis concludes GRU model for precipitation predictions from time-series based climatological data as a mechanism to precaution flood. When the precipitation forecast exceeds the threshold of 64.5 mm/day as mentioned in [8-10], it is a flood alarm for planning the disaster. This work can be extended to prediction models that combine multivariate datasets, data from multiple sources for improved precipitation forecasts.

Acknowledgment

The data was downloaded from the website https://power.larc.nasa.gov [14]. The website associates with the NASA Earth Science/Applied Science Programme and NASA Langley Research Centre (LaRC) POWER Project.

5. REFERENCES:

[1] S. N. Jonkman, "Global perspectives on loss of human life caused by floods", Natural Hazards, Vol. 34, No. 2, 2005, pp. 151-175.

[2] O. Singh, M. Kumar, "Flood occurrences, damages, and management challenges in India: a geographical perspective", Arabian Journal of Geosciences, Vol. 10, No. 102, 2017, pp. 1-19.

[3] I. Yucel, A. Onen, K.K. Yilmaz, D.J. Gochis, "Calibration and evaluation of a flood forecasting system: Utility of numerical weather prediction model, data assimilation and satellite-based rainfall", Journal of Hydrology, Vol. 523, 2015, pp. 49-66.

[4] O. Singh, M. Kumar, "Flood events, fatalities and damages in India from 1978 to 2006", Natural hazards, Vol. 69, 2013, pp. 1815-1834.

[5] Central Water Commission, Ministry of Jal shakti, Department of Water Resources, River Development and Ganga Rejuvenation, https://cwc.gov.in/flood-damage-statistics-statewise-and-country-whole-during-1953-2020 (accessed: 2023)

[6] Government of India - India Meteorological Department, https://mausam.imd.gov.in/Forecast /marquee_data/Statement_climate_of_india_2022_final.pdf (accessed: 2023)

[7] A. S. Abdullah, B. N. Ruchjana, I. G. N. M. Jaya, Soemartini, "Comparison of SARIMA and SVM model for rainfall forecasting in Bogor city", Journal of Physics: Conference Series of ICW-HDDA-X 2020, Vol. 1722, No. 1, 2021, pp. 1-8.

[8] P. Guhathakurta, O. P. Sreejith, P. A. Menon, "Impact of climate change on extreme rainfall events and flood risk in India", Journal of Earth System Science, Vol. 120, No. 3, 2011, pp. 359-373.

[9] I. G. Tunas, H. Azikin, G. M. Oka, "Impact of Extreme Rainfall on Flood Hydrographs", Proceedings of the 2nd International Conference on Hazard Mitigation in Geographic and Education Perspectives, Indonesia, 11-12 September 2020, pp. 1-6.

[10] M. Rajeevan, J. Bhate, A. K. Jaswal, "Analysis of variability and trends of extreme rainfall events over India using 104 years of gridded daily rainfall data", Geophysical Research Letters, Vol. 35, No. 18, 2008, pp.1-6.

[11] L. J. Bracken, N. J. Cox, J. Shannon, "The relationship between rainfall inputs and flood generation in south–east Spain", Hydrological Processes: An International Journal, Vol. 22, No. 5, 2008, pp. 683-696.

[12] Wikipedia, "South India", https://en.wikipedia.org/wiki/South_India (accessed: 2023)

[13] J. Subha, S. Saudia, "An Exploratory Data Analysis on SDMR Dataset to Identify Flood-Prone Months in the Regional Meteorological Subdivisions", Data Intelligence and Cognitive Informatics: Proceedings of ICDICI 2022, India, 6-7 July 2022, pp. 595-617.

[14] NASA, "Prediction of Worldwide Energy Resources", https://power.larc.nasa.gov/ (accessed: 2023)

[15] J. T. de Castro, G. M. Salistre Jr, Y. C. Byun, B. D. Gerardo, "Flashflood prediction model based on multiple regression analysis for decision support system", Proceedings of the World Congress on Engineering and Computer Science, San Francisco, CA, USA, 23-25 October 2013, pp. 23-28.

[16] E. Jumin, F. B. Basaruddin, Y. B. M. Yusoff, S. D. Latif, A. N. Ahmed, "Solar radiation prediction using boosted decision tree regression model: A case study in Malaysia", Environmental Science and Pollution Research, Vol. 28, 2021, pp. 26571-26583.

[17] J. Subha, S. Saudia, "Integrating Regression Models and Climatological Data for Improved Precipitation Forecast in Southern India", International Journal of Advanced Computer Science and Applications, Vol. 14, No. 5, 2023, pp. 626-638.

[18] A. U. Azmi, A. F. Hadi, D. Anggraeni, A. Riski, "Naive bayes methods for precipitation prediction classification in Banyuwangi", Journal of Physics: Conference Series, Vol. 1872, No. 1, 2021, pp. 1-8.

[19] Y. Dash, S. K. Mishra, B. K. Panigrahi, "Rainfall prediction for the Kerala state of India using artificial

intelligence approaches", Computers and Electrical Engineering, Vol. 70, 2018, pp. 66-73.

[20] M. Rezaeianzadeh, H. Tabari, A. Arabi Yazdi, S. Isik, L Kalin, "Flood flow forecasting using ANN, ANFIS and regression models", Neural Computing and Applications, Vol. 25, No. 1, 2014, pp. 25-37.

[21] E. G. Dada, H. J. Yakubu, D. O. Oyewola, "Artificial neural network models for precipitation prediction", European Journal of Electrical Engineering and Computer Science, Vol. 5, No. 2, 2021, pp. 30-35.

[22] R. Dey, F. M. Salem, "Gate-variants of gated recurrent unit (GRU) neural networks", Proceedings of the IEEE 60th international Midwest symposium on circuits and systems, Boston, MA, USA, 6-9 August 2017, pp. 1597- 1600.

[23] H. Song, H. Choi, "Forecasting Stock Market Indices Using the Recurrent Neural Network Based Hybrid Models: CNN-LSTM, GRU-CNN, and Ensemble Models", Applied Sciences, Vol. 13, No. 7, 2023, pp. 1-26.

[24] Y. Ji, B. Gong, M. Langguth, A. Mozaffari, X. Zhi, "CLGAN: a generative adversarial network (GAN)-based video prediction model for precipitation nowcasting", Geoscientific Model Development, Vol. 16, No. 10, 2023, pp. 2737-2752.

[25] R. Venkatesh, C. Balasubramanian, M. Kaliappan, "Precipitation prediction using generative adversarial networks with convolution neural network", Soft Computing, Vol. 25, 2021, pp. 4725-4738.

[26] S. Aswin, P. Geetha, R. Vinayakumar, "Deep learning models for the prediction of precipitation", Proceedings of the International Conference on Communication and Signal Processing, Chennai, India, 3-5 April 2018, pp. 657-661.

[27] D. Z. Haq, D. C. R. Novitasari, A. Hamid, N. Ulinnuha, Arnita, Y. Farida, R. R. D. Nugraheni, R. Nariswari, Illham, H. Rohayani, R. Pramulya, A. Widjayanto, "Long short-term memory algorithm for precipitation prediction based on El-Nino and IOD data", Proceedings of 5th International Conference on Computer Science and Computational Intelligence 2020, Indonesia, 19-20 November 2020, pp. 829-837.

[28] D. Sun, J. Wu, H. Huang, R. Wang, F. Liang, H. Xinhua, "Prediction of short-time precipitation based on deep learning", Mathematical Problems in Engineering, Vol. 2021, 2021, pp. 1-8.

[29] Y. O. Ouma, R. Cheruyot, A. N. Wachera, "Precipitation and runoff time-series trend analysis using LSTM recurrent neural network and wavelet neural network with satellite-based meteorological data: case study of Nzoia hydrologic basin", Complex & Intelligent Systems, Vol. 8, 2022, pp. 213-236.

[30] A. Samad, Bhagyanidhi, V. Gautam, P. Jain, Sangeetha, K. Sarkar, "An approach for precipitation prediction using long short term memory neural network", Proceedings of the IEEE 5th International Conference on Computing Communication and Automation, Greater Noida, India, 30-31 October 2020, pp. 190-195.

[31] M. Saha, P. Mitra, R. S. Nanjundiah, "Deep learning for predicting the monsoon over the homogeneous regions of India", Journal of Earth System Science, Vol. 126, No. 54, 2017, pp. 1-18.

[32] W. Suparta, A. A. Samah, "Rainfall prediction by using ANFIS times series technique in South Tangerang, Indonesia", Geodesy and Geodynamics, Vol. 11, No. 6, 2020, pp. 411-417.

[33] L. Zhang, R. Wang, Z. Li, J. Li, Y. Ge, S. Wa, S. Huang, C Lv, "Time-Series Neural Network: A High-Accuracy Time-Series Forecasting Method Based on Kernel Filter and Time Attention", Information, Vol. 14, No. 9, 2023, p. 500.

[34] I. V. Necesito, D. Kim, Y. H. Bae, K. Kim, S. Kim, H. S. Kim, "Deep Learning-Based Univariate Prediction of Daily Rainfall: Application to a Flood-Prone, Data-Deficient Country", Atmosphere, Vol. 14, No. 4, 2023, p. 632.

[35] B. Rekabdar, D. L. Albright, J. T. McDaniel, S. Talafha, H. Jeong, "From machine learning to deep learning: A comprehensive study of alcohol and drug use disorder", Healthcare Analytics, Vol. 2, 2022, pp. 1-18.

[36] C. Fan, M. Chen, X. Wang, J. Wang, B. Huang, "A review on data preprocessing techniques toward efficient and reliable knowledge discovery from building operational data", Frontiers in Energy Research, Vol. 9, 2021, pp. 1-17.

[37] F. Qayyum, D. H. Kim, S. J. Bong, S. Y. Chi, Y. H. Choi, “A Survey of Datasets, Preprocessing, Modeling Mechanisms, and Simulation Tools Based on AI for Material Analysis and Discovery”, Materials, Vol. 15, No. 4, 2022, p. 1428.

[38] P. Li, J. Zhang, P. Krebs, "Prediction of flow based on a CNN-LSTM combined deep learning approach", Water, Vol. 14, No. 993, 2022, pp. 1-13.

[39] H. Bohan, B. Yun, "Traffic flow prediction based on BRNN", Proceedings of the IEEE 9th International Conference on Electronics Information and Emergency Communication, Beijing, China, 12-14 July 2019, pp. 320-323.

[40] A. Dalli, "Impact of hyperparameters on Deep Learning model for customer churn prediction in telecommunication sector", Mathematical Problems in Engineering, Vol. 2022, 2022, pp. 1-11.

[41] V. Linardos, M. Drakaki, P. Tzionas, Y. L. Karnavas, "Machine learning in disaster management: recent developments in methods and applications", Machine Learning and Knowledge Extraction, Vol. 4, No. 2, 2022, pp. 446-473.

[42] J. Siłka, M. Wieczorek, M. Woźniak, "Recurrent neural network model for high- speed train vibration prediction from time series", Neural Computing and Applications, Vol. 34, No. 16, 2022, pp. 13305-13318.

[43] R. Odegua, "An empirical study of ensemble techniques (bagging, boosting and stacking)", Proceedings of the Conference Deep Learn, 2019, pp. 1-10.

[44] Z. H. Kilimci, "Ensemble Regression-Based Gold Price (XAU/USD) Prediction", Journal of Emerging Computer Technologies, Vol. 2, No. 1, 2022, pp. 7-12.

[45] S. Sankaranarayanan, M. Prabhakar, S. Satish, P. Jain, A. Ramprasad, A. Krishnan, “Flood prediction based on weather parameters using deep learning”, Journal of Water and Climate Change, Vol. 11, No. 4, 2020, pp. 1766-1783.

[46] M. N. Halgamuge, E. Daminda, A. Nirmalathas, "Best optimizer selection for predicting bushfire occurrences using deep learning", Natural Hazards, Vol. 103, No. 1, 2020, pp. 845-860.

[47] D. Chicco, M. J. Warrens, G. Jurman, “The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation”, PeerJ Computer Science, Vol. 7, 2021, pp. 1-24.

CCZO Residual GhostNet: Parkinson Disease Classification using Optimized Deep Learning Technique

Original Scientific Paper

Arogia Victor Paul M
Research Scholar, Department of Computer Science and Engineering,
B.S. Abdur Rahman Crescent Institute of Science and Technology,
Chennai, India.
victorpaul_cse@crescent.education

Sharmila Sankar
Professor & Dean, Department of Computer Science and Engineering,
B.S. Abdur Rahman Crescent Institute of Science and Technology,
Chennai, India.
sharmilasankar@crescent.education

Abstract *– Parkinson's disease (PD) classification plays a crucial role in medical diagnosis and patient management. Identifying Parkinson's disease at an early stage can lead to more effective treatment and improved patient outcomes. However, existing methods for Parkinson's disease classification face several limitations. The foremost limitation is the need for accurate and reliable diagnostic tools, as misdiagnosis can lead to inappropriate treatments and unnecessary stress for patients. Thus, a hybrid deep learning model is introduced in this research. The proposed model involves the utilization of EEG signals obtained from a publicly available dataset. Key features are extracted from the EEG signals using a bandpass filter, and every feature is associated with specific brainwave frequencies and cognitive states. The feature mapping and classification are executed through the Chaotic Chebyshev Zebra optimization-based Residual GhostNet (CCZO_Residual_GhostNet). This hybrid classifier, Residual GhostNet, combines ResNet-152 with GhostNet, enhancing classification precision. Furthermore, the CCZO algorithm optimizes the loss function, introducing elements of chaos and Chebyshev mapping to improve classification accuracy. The assessment based on accuracy, sensitivity, specificity, and F-score acquired 98.76%, 98.59%, 98.95%, and 99%, respectively.*

***Keywords**: EEG signal, hybrid deep learning, Parkinson's disease classification, Feature extraction, GhostNet, ResNet-152.*

Received: December 19, 2023; Received in revised form: January 29, 2024; Accepted: January 30, 2024

1. INTRODUCTION

People get older, and their neurons decline along with a decrease in the connections between brain cells. Nerve cells cannot replenish themselves, in contrast to other cell types in the body [1]. Neurons can get damaged or degenerate over time. Neurodegenerative illnesses are a group of disorders characterized by the progressive degeneration of the structure and function of the nervous system. These conditions often result in the gradual loss of cognitive and motor functions. Some common neurodegenerative illnesses include Alzheimer's disease, Parkinson's disease, Huntington's disease, and Amyotrophic Lateral Sclerosis (ALS). Parkinson's disease (PD) is a neurodegenerative illness that primarily affects neurons in the brain's substantia nigra. These neurons are essential for the synthesis of dopamine, which is a neurotransmitter that connects neurons in the brain [2]. Dopamine helps messages go from the brain to other regions of the body, especially when it comes to speech articulation and physical motions. When a considerable proportion of dopaminergic neurons degenerate or when dopamine levels in the brain diverge from normal, Parkinson's disease symptoms become apparent [3]. Statistics from the World Health Organization indicate that about 10 million people suffer from the effects of this illness. It is more common in older adults, affecting those in their fifties and older. Males are 1.5 times more prone to PD than females, and around 4% of cases are identified before the age of fifty [4]. The initial symptoms could be difficult to notice and modest at first, but they get worse over time. Dyskinesia, syncope, exhaustion, tremors, stiffness, dystonia, hypomimia, diarrhea, poor smell or taste, and loss of weight are examples of both motorized and non-motorized symptoms. Because PD is untreatable, early diagnosis is essential for patients to take proactive steps for managing the condition, which allows them to continue with their regular activities [5].

Depending on how Parkinsonism is classified, many imaging modalities are used to diagnose PD [6]. There are different kinds of Parkinsonism, and this study focused on the most common kind, idiopathic PD, usually referred to as PD, which has an unclear etiology [7]. As part of the diagnostic procedure for PD, PET (positron emission tomography) and SPECT (single photon emission computed tomography) show exceptional sensitivity in detecting dopamine shortages [8]. However, the high cost and specialized equipment required for these imaging modalities limit their broad use in routine clinical diagnosis [9]. In addition to imaging techniques, 90% of patients who undergo the olfactory dysfunction test are utilized as a preliminary clinical sign of PD. Techniques based on biomarkers include quantifying biological markers found in different parts of the body and blood to provide information on the existence and severity of illness. Another potential diagnostic method for PD is electroencephalography (EEG) [10]. EEG-based treatments have several benefits with respect to other diagnostic techniques, such as cost-effectiveness, non-interfering, and better resolution, as they are non-invasive. The number of studies utilizing EEG technology is increasing [11, 12].

Many different methods are presented in this field; most of them use speech signals, handwriting signals, gait signals, MRI, and very few use EEG. One of the most effective methods for diagnosing PD is electroencephalography (EEG) [13]. Since EEG technology is portable and affordable, its value is demonstrated by its capacity to record brain activity in real-world settings [14]. Moreover, EEG-record-based brain activity occurs faster than other modalities and for longer periods. Thus, the analysis of EEG integrated with machine learning techniques has already proven to be useful in the diagnosis of a number of neurological disorders, including epilepsy, major depressive disorder, schizophrenia, Alzheimer's disease, autism spectrum disorder, and dementia [15, 16]. The amount of medical data that is being recorded has grown to incredible heights; signals and photographs in particular have amassed gigabytes and even terabytes of data. It is a laborious undertaking to process these enormous datasets and extract valuable insights from them. One aspect of artificial intelligence called machine learning gives machines the ability to anticipate outcomes based on data analysis, teaching them to mimic human intellect [17]. Thus, a novel deep learning-based framework is introduced in this research. The major contributions of the research are:

- Design of CCZO Algorithm: The proposed CCZO algorithm is designed by integrating the chaotic Chebyshev mapping with zebra optimization to enhance the randomization criteria for obtaining the global best solution.
- Design of hybrid Residual_GhostNet: The hybrid deep learning by integrating the ResNet-152 with the GhostNet to improve the classification accuracy.
- Design of CCZO-Residual_GhostNet for PD classification: The PD classification is employed using hybrid Residual_GhostNet, wherein the loss function optimization is employed using the CCZO algorithm.

The organization of the research is: Section 2 details the related works and Section 3 explains the Proposed PD classification. Section 4 elaborates the experimental outcome and Section 5 concludes the research.

2. RELATED WORKS

This section offers a survey of the literature on machine learning-based Parkinson disease classification. The EEG signal was used by [18] to distinguish between individuals with PD who were taking medication and those who were not. Pre-processing of the signals was done in order to remove significant artifacts. Based on the collected characteristics, [19] created a collection of machine learning methods for classifying Parkinson's illness. These methods make it possible to automatically classify EEG data into those with PD and those without it. In this case, the discriminative characteristics of Parkinson's illness were improved by the use of spatial filtering. Analyzing variables such as frequency bands, segment lengths, and feature reduction numbers provides valuable information for improving the suggested approaches' efficiency and versatility. Complexity may be introduced by utilizing various machine learning algorithms and feature extraction metrics, which can make the models difficult to comprehend and use in clinical contexts. To accurately classify PSD and healthy control (HC) participants, a convolutional neural network (CNN)-based classification model with seven hidden layers and various filter sizes was suggested [20]. Three-dimensional data was transformed into a one-dimensional tensor flow using a flattening layer. In order to determine the initial danger of PSD patients, the dense layer finally outputs a categorization of HC and PSD patients depending on the strength of their tremors. With a tremor detection rate of 92.4%, it surpassed the conventional models. In order to demonstrate the value of deep learning-driven voice recognition as a diagnostic instrument for Parkinson's disease (PD), a speech signal processing technique was suggested [21]. It was explored if voice recordings could offer a straightforward, inexpensive approach to assessing and testing for Parkinson's disease, utilizing deep learning to forecast and assess expert scores. As a result, a modified Hybrid Mask U-Net architecture with an adaptive custom loss function called the Deep U-lossian model was developed for PD assessment and recognition, aiming toward an improved ratio of recall and precision in handled speech.

It is discussed how to classify the high-dimensional PD data [22]. In order to create effective ML classifiers to classify Parkinson's disease (PD), the best subset of features from the PD data set is chosen using a bio-inspired feature selection strategy. Eleven machine learning classifiers (ML) were used in the study: LR, lSVM, rSVM, GNB, GPC, kNN, DT, RF, MLP, AB, and QDC. Two bioinformatics techniques (GA and BPSO) were used for feature selection. The PD data set is split into training and testing sets in the ratio of 0.7:0.3 to train and test all 11 ML classifiers. Based on numerous classification assessment measures, the effectiveness of these ML classifiers is assessed both prior to and following the selection of bioinspired features. The presented results indicate that three of the best BPSO-inspired classifiers, BPSOMLP, and three of the best GA-inspired classifiers, GAMLP, GAGPC, and GALR, can be suggested for categorizing the PD data.

2.1. PROBLEM STATEMENT

PD is a crippling neurological ailment that has several negative consequences. It mostly affects the motor function of the person, resulting in symptoms like tremors, muscular stiffness, and postural instability. As the illness worsens, mobility issues may arise, increasing the risk of falls and associated injuries. PD can also include non-motor symptoms such as anxiety, sadness, insomnia, and cognitive decline. Difficulties with swallowing and speech might also occur, making everyday living even more challenging. PD can have a significant emotional and social impact on a person, sometimes resulting in social disengagement, a decline in daily functioning, and a breakdown of relationships.

Various methods are currently used for PD diagnosis, but they come with their own set of challenges. Imaging techniques like MRI and DaTscan can visualize brain changes, but they are costly and not always readily available. Biosensors offer continuous monitoring but struggle to distinguish PD from other movement disorders. Genetic testing can identify rare mutations linked to PD, but most cases do not involve these mutations. EEG-based methods can detect brain activity changes but require advanced data analysis and interpretation.

The hybrid Residual_GhostNet model represents a promising approach to overcome these challenges. By using deep learning and neural networks, this model can analyze EEG data, identifying patterns associated with PD more objectively and efficiently. It leverages a data-driven approach for automatic extraction of relevant features from EEG signals, reducing the need for manual feature engineering and human interpretation. Loss function optimization technique using CCOZ fine-tunes the model's parameters, enhancing its ability to classify PD accurately. With the automation and efficiency of deep learning models, the Hybrid Residual_GhostNet can analyze large datasets rapidly, offering a potential solution to the challenges of subjectivity in clinical assessments, early-stage PD detection, and the requirement for cost-effective and non-invasive diagnostic tools. This model represents a significant advancement in the field of Parkinson's disease detection, potentially leading to more timely diagnoses and improved patient care.

3. PROPOSED METHODOLOGY

The proposed PD classification is presented in Fig. 1, wherein the input EEG signal is acquired from the publicly available dataset. Initially, the essential features are extracted from the EEG signal by the bandpass filter. From the extracted features, feature mapping and classification are employed using the proposed Chaotic Chebyshev Zebra optimization-based Residual GhostNet (CCZO_Residual_GhostNet). Here, the hybrid classifier Residual GhostNet is designed by integrating ResNet-152 with GhostNet. Besides, the loss function optimization is employed using the CCZO algorithm designed by incorporating the chaotic Chebyshev with the conventional zebra optimization algorithm for enhancing the classification accuracy.

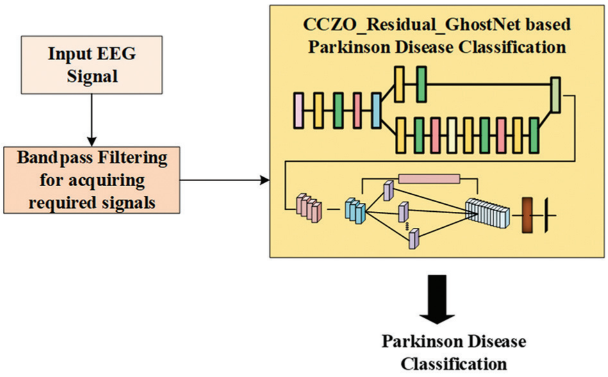

Input EEG Signal
Bandpass Filtering for acquiring required signals
CCZO_Residual_GhostNet based Parkinson Disease Classification
Parkinson Disease Classification

Fig. 1. Workflow of proposed PD classification

3.1. DATA ACQUISITION

The input data for processing the PD classification is acquired from the publically available dataset named the UCSD dataset.

3.2. FEATURE EXTRACTION

The acquired EEG signal is filtered using the bandpass filter to acquire the required features alpha, beta, gamma, delta, and theta.

Delta Waves (0.5-4 Hz): Delta oscillations manifest as the most languid cerebral frequencies, intimately entwined with profound slumber, tranquility, and states of subliminal awareness. They organize the symphony of physical and psychological rejuvenation.

Theta Waves (4-8 Hz): Theta waves find their similarity with profound serenity, dream, and the primary phases of inactivity. These waves are also patrons of ingenuity and transcendental contemplation.

Alpha Waves (8-13 Hz): Alpha rhythms take centre stage when one is in an awakened yet tranquil disposition.

They often grace us when our eyelids are shut, heralding a composed and vigilant mentality.

Beta Waves (13-30 Hz): Beta frequencies accompany lively, conscious cogitation and acumen. They flourish during periods of vigilance, attentiveness, and cognitive riddles.

Gamma Waves (30-100 Hz and beyond): Gamma waves, the swiftest of neural harmonics, intertwine with loftier cognitive faculties, perception, and cognizance. They partake in the orchestration of informational processing and may be linked with epiphanies.

From the acquired signal, the PD classification is employed.

3.3. PD CLASSIFICATION USING IMPROVED RESIDUAL_GHOSTNET

The Parkinson's disease classification is employed using the proposed Improved Residual_GhostNet. Here, the ResNet-152 is integrated with the GhostNetto enhance the disease classification precision. Besides, the loss function optimization is devised using the CCZO algorithm for enhancing the classification accuracy further.

3.3.1. Architecture of ResNet

ResNet architectures are known for their skip connections, also called residual connections. These connections enable the network to skip over one or more layers and add the output from a previous layer to the output of a subsequent layer. This is done through element-wise addition. The key criteria for a skip connection are that the dimensions of the feature maps must match. ResNet-152 is a deep convolutional neural network with 152 layers. It comprises various components, including convolutional layers, residual blocks, maxpooling, fully connected layers, and activation functions. These components work together to map the features from the input features to perform the classification more accurately. The architecture of the ResNet-152 is depicted in Fig. 2.

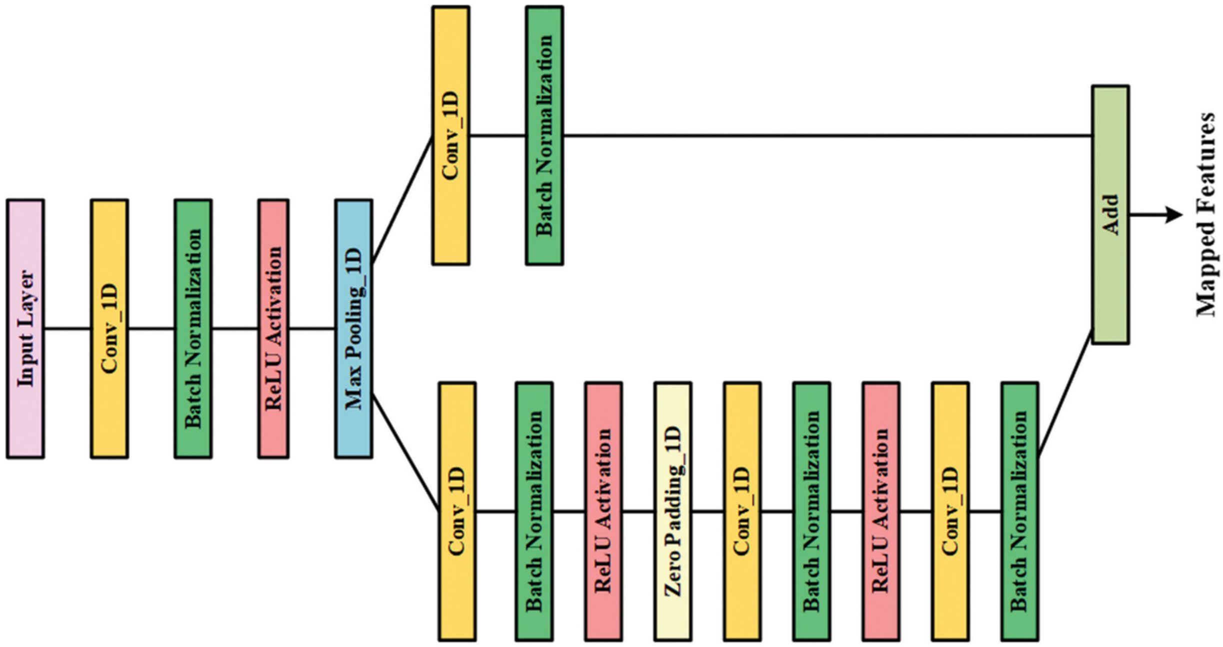


Fig. 2. Architecture of ResNet-152

3.3.2. Architecture of GhostNet

Using the features mapped by ResNet-152, the PD classification is employed using GhostNet. The utilization of GhostNet for PD classification offers a range of significant benefits. GhostNet, renowned for its efficiency and compact architecture, stands out as an optimal choice in the field of disease diagnosis and classification. Its lightweight design and reduced computational requirements result in faster inference times, making it ideal for real-time or near-real-time applications. This attribute is particularly crucial in the context of healthcare, where swift diagnosis and monitoring are paramount. The architecture of GhostNet is depicted in Fig. 3.

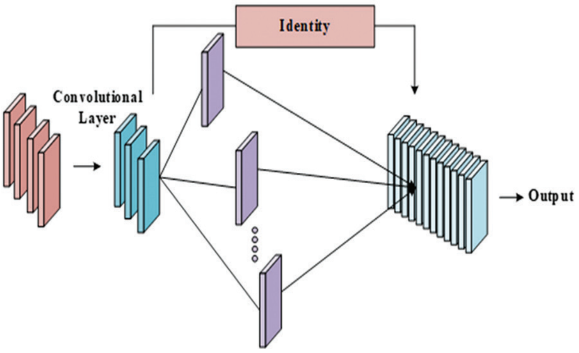


Fig. 3. Architecture of GhostNet

The two various paths utilized by the GhostNet are the:

Convolutional Layer: The Convolutional Path is the primary pathway in GhostNet for processing input data. It consists of standard convolutional layers, which are fundamental in deep learning for feature extraction. The Convolutional Path plays a critical role in capturing low- and high-level features from the data, gradually building a hierarchical representation of the input. The outcome of the Convolutional layer is represented as:

$$CL = D * b + f \tag{1}$$

where, refers the outcome of the convolutional layer, bias is notated as, the input data is represented as, and defines the conventional filters.

Ghost Layer: The Ghost Path is a distinctive aspect of GhostNet's architecture. It complements the Convolutional Path to improve feature representation and model performance. The Ghost Path consists of ghost modules, which are essentially lightweight versions of standard convolutional layers. These ghost modules are created by using depth-wise separable convolutions. In the Ghost Path, the ghost modules are designed to capture additional features and patterns in the input data. They operate in parallel with the Convolutional Path.

$$GL = D * f' \tag{2}$$

where, the filter utilized in the ghost path is denoted as and the outcome of the Ghost module is defined as.

The outputs from the ghost modules are then combined with the outputs from the Convolutional Path. This fusion of information enhances the network's ability to learn discriminative features while maintaining efficiency.

3.3.3. Architecture of Residual_ GhostNet

The proposed Residual_GhostNet is designed by integrating the conventional ResNet-152 with the GhostNet for enhancing the classification accuracy, which is depicted in Fig. 4. In this the outcome of the GhostNet is connected with the fully connected layer and the softmax layer for classifying the PD.

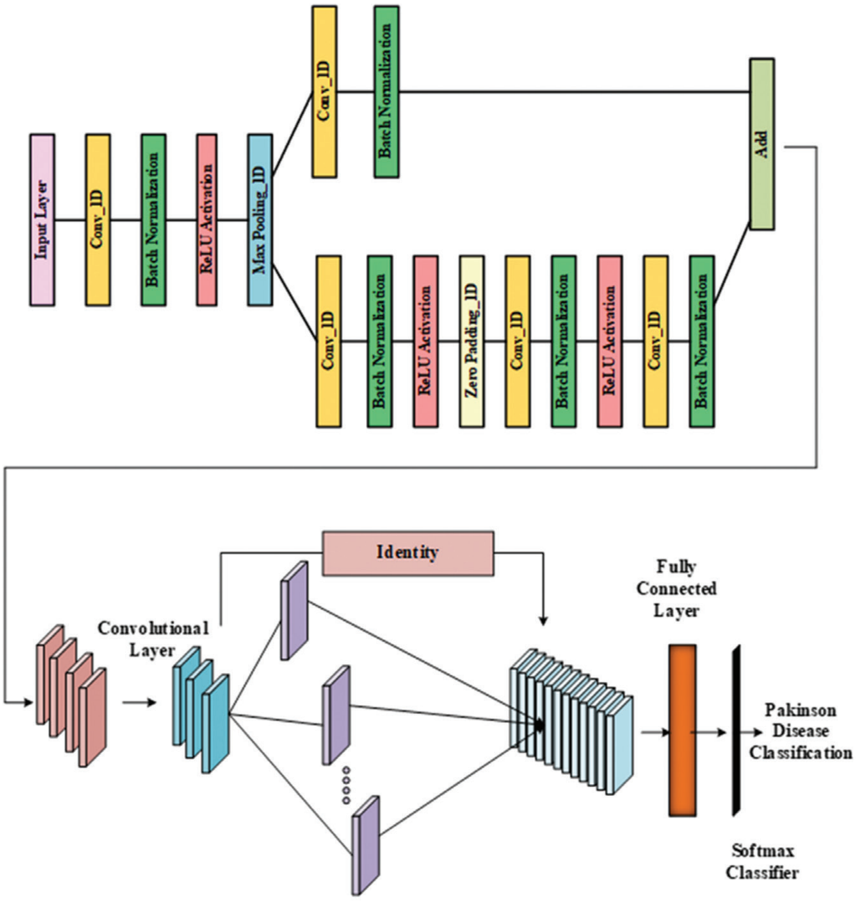


Fig. 4. Architecture of Residual_GhostNet

Here, the proposed PD model is tuned optimally using the CCZO algorithm for enhancing classification accuracy.

Loss Function Optimization: The loss function optimization is devised using the proposed Chaotic Chebyshev Zebra Optimization (CCZO) algorithm. In this, the solution trapping at the local optima is eliminated by incorporating the randomness criteria in the exploration phase using the Chaotic Chebyshev mapping.

Initialization: Each zebra in this population is like a potential solution to a problem that the algorithm is trying to solve. The location of each zebra on the search space represents a set of values for the decision variables related to the problem. Essentially, the zebra's positions correspond to different potential solutions. This randomness is part of the algorithm's exploration of several solutions. The initialization of the population is stated as:

$$A = \begin{bmatrix} A_1 \\ \vdots \\ A_k \\ \vdots \\ A_G \end{bmatrix} = \begin{bmatrix} a_{1,1} & \cdots & a_{1,l} & \cdots & a_{1,b} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ a_{k,1} & \cdots & a_{k,l} & \cdots & a_{k,b} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ a_{G,1} & \cdots & a_{G,l} & \cdots & a_{G,b} \end{bmatrix}_{G\times b} \tag{3}$$

Here, the population of the zebra is denoted as A, and A_k refers to the k^{th} zebra in the search space. The total count of zebras considered in the algorithm is denoted as G, and $a_{k,\,l}$ represents the k^{th} zebra with the solution dimension l. After placing the population, the feasibility of the solution is evaluated.

Feasibility Evaluation: The feasibility of the solution is estimated for every zebra to identify the closeness of the solution to the required target. In the proposed disease detection, the mean square error is considered for evaluating the feasibility and is stated as:

$$F = \frac{1}{T}\sum_{x=1}^{T}(O_x - T_x)^2 \tag{4}$$

where, the fitness is F, overall samples is denoted as T, the observed value is O_x and the target value is denoted as T_x.

Randomization: The randomization of the algorithm utilizes the foraging behaviour, wherein the food searching is employed. A specific type of zebra grasses in the plains and is named as Pioneer zebra that leads the group members to get the food and updates the solution as:

$$a_{k,l}^{R} = a_{k,l} + q \cdot \left(DE_l - H \cdot a_{k,l}\right) \tag{5}$$

Here, the zebra that guides the team members in obtaining the food is denoted as DE_l and the value [0,1] is the limit for the arbitrarily chosen variableq. The expression for identifying the factor H is expressed as:

$$H = round(1 + d) \tag{6}$$

Here, random number denoted as dhas the limit [0,1] and hence, the range of the factor H is varies from {1,2}. After evaluating the solution for the zebras, the updation of the acquired solution is devised by:

$$A_k = \begin{cases} A_k^R F_k^R < F_k \\ A_k Otherwise \end{cases} \tag{7}$$

The solution accomplished by the zebra in the randomization phase is denoted as $a_{k,l}^{R}$, and the fitness for this phase is defined as F_k^R.

Here, in the randomization phase, the chaotic chebychev randomization is incorporated with the foraging behaviour of the zebra for enhancing the exploration strategy to obtain the global best solution. The expression that represents the chaotic chebyshev randomization is expressed as:

$$A_k^R = cos(J.cos^{-1} A_k) \tag{8}$$

$$A_k^R = 0.5[A_k^R]_{Zebra} + 0.5[A_k^R]_{chaoticchebyshev} \tag{9}$$

$$A_k^R = 0.5\left[a_{k,l} + q \cdot \left(DE_l - H \cdot a_{k,l}\right)\right] + 0.5[cos(J.cos^{-1} A_k)] \tag{10}$$

Thus, using the equation (10), the solution updation is devised using the CCZO algorithm and assist to obtain the global best solution.

Escaping Capability: In this phase, the zebra tries to escape from the predator like the lion. Similarly, zebras offend some predators like dogs and hyena's. Thus, the solution updation devised by the zebra in both the escaping and offending capability is expressed as:

$$a_{k,l}^{R2} = \begin{cases} a_{k,l} + Q \cdot (2q-1) \cdot \left(1 - \frac{\tau}{\tau_{max}} ()_{k,l_s}\right) \\ a_{k,l} + q \cdot \left(R_l - H \cdot a_{k,l}\right), otherwise \end{cases} \tag{11}$$

Here,

$$A_k = \begin{cases} A_k^{R2} F_k^{R2} < F_k \\ A_k Otherwise \end{cases} \tag{12}$$

Thus, using the evaluation of the fitness, the solution updation is devised.

Stoppage: The acquisition of the targeted solution or the completion of the iteration stops the iteration processing.

4. RESULT AND DISCUSSION

The implementation of the proposed PD classification is performed using the PYTHON programming language. Besides, the comparison with the conventional PD classification methods likes 2D-CNN [22], CSP+KNN [19], DWT+SVM [18] and Channelwise CNN [20] for depicting the superiority of the proposed model.

4.1. DATASET DESCRIPTION

The dataset comprises of various EEG signal, where in each EEG recording in the dataset is associated with a label indicating the presence or absence of Parkinson's disease. These labels are essential for supervised machine learning tasks, where the goal is to classify EEG signals as either Parkinson's disease or non-Parkinson's disease.

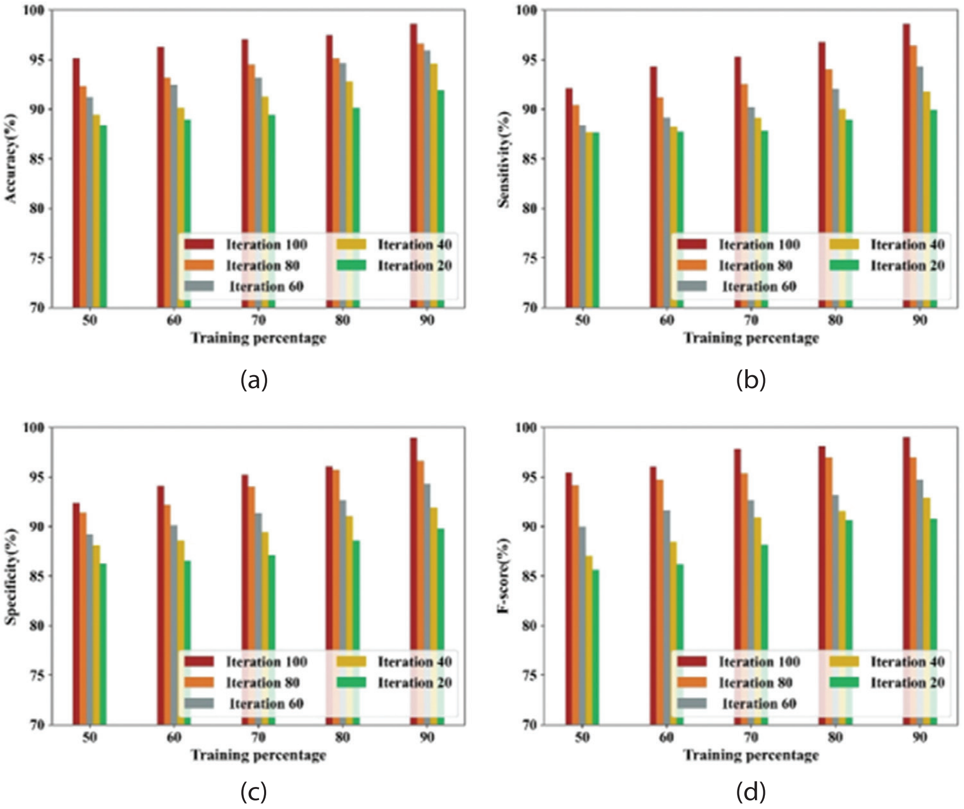
4.2. PERFORMANCE ANALYSIS

The performance evaluation of the proposed CCZO-Residual_GhostNet model for various iterations is visualized in Fig. 5. When using 100 iterations with 50% of the data allocated for training, the model achieves an accuracy of 95.12%. However, when the model is evaluated with 80, 60, 40, and 20 iterations, the accuracy decreases to 92.33%, 91.23%, 89.42%, and 88.37%, respectively. This analysis reveals that the model performs better with a higher number of iterations and a larger percentage of training data.

The superior outcomes in these scenarios are attributed to the use of the CCZO algorithm for loss function optimization. This optimization enhances the model's generalization capability, allowing it to achieve higher accuracy and better performance.

4.3. COMPARATIVE ANALYSIS

The comparative assessment of PD classification is visualized in Fig. 6. In this assessment, the accuracy achieved by the CCZO_Residual_GhostNet model is 96.01%. This accuracy outperforms the 2D-CNN, CSP+KNN, DWT+SVM, and Channelwise-CNN methods by margins of 1.40%, 2.00%, 4.10%, and 5.79%, respectively, when 80% of the data is used for training. Likewise, when considering sensitivity, the CCZO_Residual_GhostNet model demonstrates a sensitivity of 94.21%. This sensitivity surpasses the 2D-CNN, CSP+KNN, DWT+SVM, and Channelwise-CNN methods by margins of 3.16%, 5.40%, 6.00%, and 7.18%, respectively, when 70% of the data is allocated for training.



(a)

(b)

(c)

(d)

Fig. 5. Analysis of Improved Residual_GhostNet (a) accuracy, (b) Sensitivity, (c) Specificity and (d) F-Score

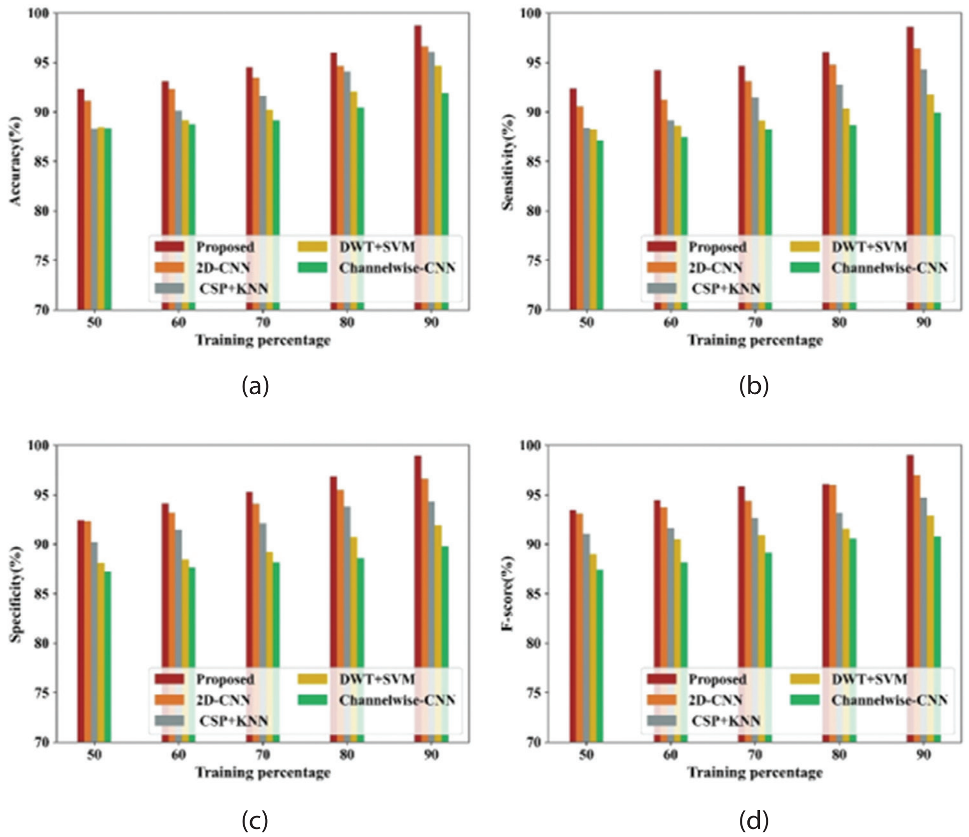
The analysis provided in the table offers insights into how the hybrid deep learning model excels in classifying the disease with minimal complexity. This efficiency is attributed to the model's minimal number of layers, which effectively capture essential features for accurate classification.

The accuracy-loss analysis of the proposed PD classification method is presented in Fig. 7. The accuracy analysis depicts the superior outcome for the training data compared to the testing data. Similarly, the loss function is higher for the testing data. But the performance is closer to the training data.

4.4. COMPARATIVE DISCUSSION

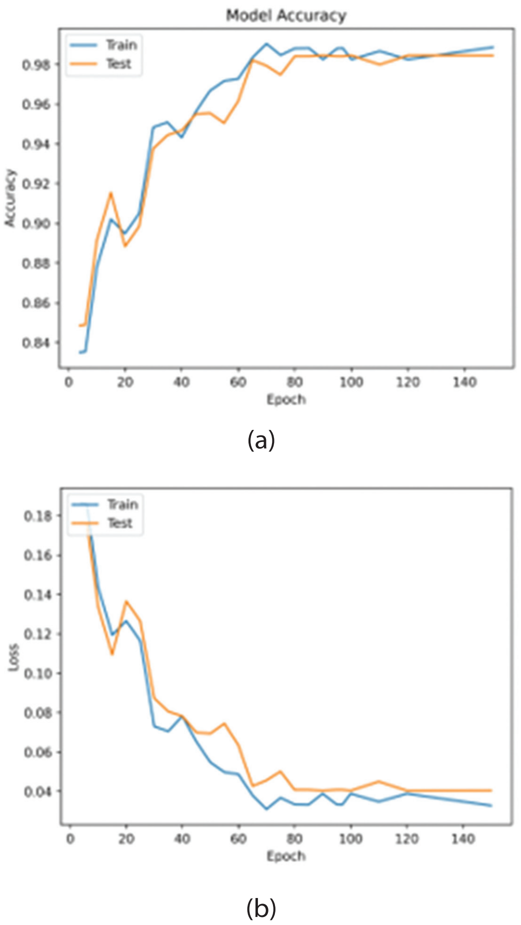
The precision ascertained through the adept CCZO_Residual_GhostNet achieves a remarkable 98.76%, bestowing a substantial superiority of 2.18%, 2.74%, 4.13%, and 6.96% in contrast to the 2D-CNN, CSP+KNN, DWT+SVM, and Channelwise-CNN techniques. Further delving into the assessment, the sensitivity estimations courtesy of the CCZO_Residual_GhostNet reveal a remarkable edge. The metrics stand at 98.59%, eclipsing their counterparts by margins of 2.20%, 4.37%, 6.94%, and 8.77% when juxtaposed with 2D-CNN, CSP+KNN, DWT+SVM, and Channelwise-CNN, respectively. It is paramount to elucidate the specificity aspect, where the CCZO_Residual_GhostNet truly excels. Recording an estimable score of 98.95%, it soars above the 2D-CNN, CSP+KNN, DWT+SVM, and Channelwise-CNN by significant differentials of 2.32%, 4.68%, 7.13%, and 9.26%. To further enhance the narrative, the F-Score, a comprehensive metric of precision and recall, is a standout. The CCZO_Residual_GhostNet attains an impressive 99%, accentuating its dominance over its peers. These achievements underscore the model's superiority, boasting advantages of 2.02%, 4.31%, 6.17%, and 8.29% compared to the 2D-CNN, CSP+KNN, DWT+SVM, and channelwise-CNN methods, respectively.

Here, the analysis depicts the superior outcome in terms of all assessment measures by the proposed model. Several real-world applications of Parkinson's disease classification using deep learning are emerging, showing promise for improving diagnosis, treatment, and patient care.



(a) (b) (c) (d)

Fig. 6. Comparative Analysis (a) accuracy, (b) Sensitivity, (c) Specificity and (d) F-Score



(a)

(b)

Fig. 7. Accuracy-Loss Analysis: (a) Accuracy and (b) Loss

5. CONCLUSION

In summary, this research presents a robust method for PD classification. By employing the CCZO_Residual_GhostNet model, we achieve superior accuracy, sensitivity, specificity, and F-Score compared to conventional methods such as 2D-CNN, CSP+KNN, DWT+SVM, and Channelwise-CNN. The utilization of ResNet-152 with skip connections, coupled with GhostNet's efficient architecture, ensures an efficient tool for disease classification. The incorporation of the CCZO algorithm further refines the model's performance, eliminating local optima and enhancing global optimization. This method offers a promising approach for accurate and efficient PD classification using EEG signals, contributing to advancements in the field of medical diagnosis and treatment. In the future, it might be helpful to recognize present clinical data sets that have been utilized that could help the clinical classification of the illness, like DaTscan, or to capitalize on information set methods such as sleep EEGs, which could help with the possible rapid detection of biological indicators of PD and its associated issues, such as MCI and dementia.

6. REFERENCES:

[1] A. Smrdel, "Use of common spatial patterns for early detection of Parkinson's disease", Scientific Reports, Vol. 12, No. 1, 2022, p. 18793.

[2] M. Shaban, "Deep Learning for Parkinson's Disease Diagnosis: A Short Survey", Computers, Vol. 12, No. 3, 2023, p. 58.

[3] Y. Yang, Y. Yuan, G. Zhang, H. Wang, Y. C. Chen, Y. Liu, D. Katabi, "Artificial intelligence-enabled detection and assessment of Parkinson's disease using nocturnal breathing signals", Nature Medicine, Vol. 28, No. 10, 2022, pp. 2207-2215.

[4] A. A. Bhurane, S. Dhok, M. Sharma, R. Yuvaraj, M. Murugappan, U. R. Acharya, "Diagnosis of Parkinson's disease from electroencephalography signals using linear and self-similarity features", Expert Systems, Vol. 39, No. 7, 2022, p. e12472.

[5] A. M. Maitin, J. P. R. Muñoz, Á. J. García-Tejedor, "Survey of machine learning techniques in the analysis of EEG signals for Parkinson's disease: A systematic review", Applied Sciences, Vol. 12, No. 14, 2022, p. 6967.

[6] M. Parajuli, A. W. Amara, M. Shaban, "Deep-learning detection of mild cognitive impairment from sleep electroencephalography for patients with Parkinson's disease", PLoS One, Vol. 18, No. 8, 2023, p. e0286506.

[7] I. Suuronen, A. Airola, T. Pahikkala, M. Murtojärvi, V. Kaasinen, H. Railo, "Budget-based classification of Parkinson's disease from resting state EEG", IEEE Journal of Biomedical and Health Informatics, Vol. 27, No. 8, 2023, pp. 3740-3747.

[8] M. Shaban, A. W. Amara, "Resting-state electroencephalography based deep-learning for the detection of Parkinson's disease", PLoS One, Vol. 17, No. 2, 2022, p. e0263159.

[9] L. Qiu, J. Li, J. Pan, "Parkinson's disease detection based on multi-pattern analysis and multi-scale convolutional neural networks", Frontiers in Neuroscience, Vol. 16, 2022, p. 957181.

[10] K. H. Chang, I. T. French, W. K. Liang, Y. S. Lo, Y. R. Wang, M. L. Cheng, C. H. Juan, "Evaluating the different stages of Parkinson's disease using electroencephalography with Holo-Hilbert spectral analysis", Frontiers in Aging Neuroscience, Vol. 14, 2022, p. 832637.

[11] Y. Guo, D. Huang, W. Zhang, L. Wang, Y. Li, G. Olmo, P. Chan, "High-accuracy wearable detection of freezing of gait in Parkinson's disease based on pseudo-multimodal features", Computers in Biology and Medicine, Vol. 146, 2022, p. 105629.

[12] M. A. Motin, M. Mahmud, D. J. Brown, "Detecting Parkinson's disease from electroencephalogram signals: An explainable machine learning approach", Proceedings of the IEEE 16th International Conference on Application of Information and Communication Technologies, Washington DC, DC, USA, 12-14 October 2022, pp. 1-6.

[13] S. Avvaru, K. K. Parhi, "Effective Brain Connectivity Extraction by Frequency-Domain Convergent Cross-Mapping (FDCCM) and its Application in Parkinson's Disease Classification", IEEE Transactions on Biomedical Engineering, Vol. 70, No. 8, 2023, pp. 2475-2485.

[14] L. di Biase, L. Ricci, M. L. Caminiti, P. M. Pecoraro, S. P. Carbone, V. Di Lazzaro, "Quantitative High Density EEG Brain Connectivity Evaluation in Parkinson's Disease: The Phase Locking Value (PLV)", Journal of Clinical Medicine, Vol. 12, No. 4, 2023, p. 1450.

[15] B. F. O. Coelho, A. B. R. Massaranduba, C. A. dos Santos Souza, G. G. Viana, I. Brys, R. P. Ramos, "Parkinson's disease effective biomarkers based on Hjorth features improved by machine learning", Expert Systems with Applications, Vol. 212, 2023, p. 118772.

[16] M. Nour, U. Senturk, K. Polat, "Diagnosis and classification of Parkinson's disease using ensemble learning and 1D-PDCovNN", Computers in Biology and Medicine, Vol. 161, 2023, p. 107031.

[17] R. Parameshwara, S. Narayana, M. Murugappan, R. Subramanian, I. Radwan, R. Goecke, "Automated Parkinson's Disease Detection and Affective Analysis from Emotional EEG Signals", arXiv:2202.12936, 2022.

[18] M. Aljalal, S. A. Aldosari, M. Molinas, K. AlSharabi, F. A. Alturki, "Detection of Parkinson's disease from EEG signals using discrete wavelet transform, different entropy measures, and machine learning techniques", Scientific Reports, Vol. 12, No. 1, 2022, p. 22547.

[19] A. M. Abdurraqeeb, F. A. Alturki, "Parkinson's Disease Detection from Resting-State EEG Signals Using Common Spatial Pattern, Entropy, and Machine Learning Techniques", Diagnostics, Vol. 12, No. 5, 2022, p. 1033.

[20] J. J. Hathaliya, H. Modi, R. Gupta, S. Tanwar, P. Sharma, R. Sharma, "Parkinson and essential tremor classification to identify the patient's risk based on tremor severity", Computers and Electrical Engineering, Vol. 101, 2022, p. 107946.

[21] R. Maskeliūnas, R. Damaševičius, A. Kulikajevas, E. Padervinskis, K. Pribuišis, V. Uloza, "A hybrid U-lossian deep learning network for screening and evaluating Parkinson's disease", Applied Sciences, Vol. 12, No. 22, 2022, p. 11601.

[22] A. Pasha, P. H. Latha, "Bio-inspired dimensionality reduction for Parkinson's disease (PD) classification", Health Information Science System, Vol. 8, No. 13, 2020.

Minimizing Noise in Location Privacy Protection Through Equipment Error Consideration

Original Scientific Paper

Riho Isawa
The University of Electro-Communications,
Graduate School of Informatics and Engineering
Departments, Department of Informatics
1-5-1 Chofugaoka, Chofu, Japan
isawa.riho@ohsuga.lab.uec.ac.jp

Yasuyuki Tahara
The University of Electro-Communications,
Graduate School of Informatics and Engineering
Departments, Department of Informatics
1-5-1 Chofugaoka, Chofu, Japan
tahara@uec.ac.jp

Yuichi Sei
The University of Electro-Communications,
Graduate School of Informatics and Engineering
Departments, Department of Informatics
1-5-1 Chofugaoka, Chofu, Japan
seiuny@uec.ac.jp

Akihiko Ohsuga
The University of Electro-Communications,
Graduate School of Informatics and Engineering
Departments, Department of Informatics
1-5-1 Chofugaoka, Chofu, Japan
ohsuga@uec.ac.jp

Abstract – *In recent years, systems that collect location information and publish statistics, such as those that publish congestion information, have been extensively employed. Because it is possible to infer an individual's identity even if the information is not directly disclosed, it is essential to disclose data with privacy protection. Therefore, privacy protection methods based on differential privacy are attracting attention. Geo-indistinguishability is the most famous extension theorem of differential privacy for location information. Geo-indistinguishability can be achieved by adding noise to a target value that must be protected. However, noise addition reduces the usefulness of the data. Thus, it is desirable to add minimal noise to your privacy budget. Therefore, we focus on the fact that the values obtained using measurement devices contain errors. We introduced a novel concept of differential privacy tailored for location information, termed true-value-based geo-indistinguishability ($T\text{-}Geo\text{-}I$), which accounts for equipment noise. We also proposed a location information privacy protection method that considers $T\text{-}Geo\text{-}I$ and reduces the amount of added noise. The object of privacy protection should be the "true value" not the "measured value" that includes measurement errors.*

According to the experimental results, in the case wherein the measurement error is the normal distribution, our method reduced the noise average and mean square error (MSE) by up to 41% and 63%, respectively, compared with conventional methods while maintaining a prespecified level of privacy in 10^8 samples of numerical data. In the case wherein the measurement error is the lognormal distribution, the proposed method based on $T\text{-}Geo\text{-}I$ succeeded in reducing the noise average and MSE by up to 60% and 67%, respectively, compared with methods based on $Geo\text{-}I$, while maintaining a prespecified level of privacy. These findings indicate that the proposed method can improve the usefulness of data while maintaining a prespecified degree of privacy protection.

Keywords: *differential privacy, geo-indistinguishability, GPS, Laplace mechanism, local differential privacy, location-based service*

Received: December 12, 2023; Received in revised form: February 10, 2024; Accepted: February 10, 2024

1. INTRODUCTION

In recent years, systems that collect location information and publish statistics, such as those that publish congestion information, have been extensively employed. The Internet of Things (IoT) technology has revolutionized innovation in people's lives by collecting and storing information received from physical objects or sensors [1–2]. Although these systems are convenient, they carry the risk of leaking personal information such as location information [3]. Even if personal information is not directly disclosed, it may be inferred from statistical data. Storing and using information on the cloud is also becoming more prevalent [4–5]. Location privacy preservation is essential, and there are many research challenges [6–7].

When disclosing statistical data to the public, it is essential to take privacy into account and perform processing to ensure that individuals cannot be identified from the data before releasing the data. Recently, privacy protection methods based on differential privacy have attracted attention. Representative examples of privacy protection for location information based on differential privacy include NTT Docomo's mobile spatial statistics and Google Maps processing of congested areas. Differential privacy is used in statistics in the real world and is widely recognized as a security indicator that can suppress the disclosure of data privacy, regardless of the attacker's background knowledge or attack method algorithm.

Geo-indistinguishability (*Geo-I*) is attracting attention as a standard that applies differential privacy to protect location information data [8]. It shows the guaranteed criteria when noise is added to the position information using the perturbation method on the Euclidean plane. One perturbation method that satisfies the *Geo-I* criteria and protects the true value by adding random noise to a person's location information is the planar Laplace mechanism. This method protects privacy by adding noise that satisfies the criterion of differential privacy to the true data using the Laplace distribution. In general, the stronger the degree of privacy protection, the higher the amount of noise added, which reduces the usefulness of the data. There is a trade-off between the usefulness of data and the degree of privacy protection.

Because the degree of privacy protection is specified numerically, there is a need for a noise addition method that satisfies this degree of protection in terms of differential privacy. To enhance the usefulness of the data, the amount of noise added to the true value should be reduced. The more noise added, the less useful the data becomes. Therefore, we focus on the fact that the measured values already contain errors and attempt to suppress the total amount of added noise. Because conventional methods do not consider errors during measurement, they may contain extra noise for the privacy protection parameter budget. In general, technologies for obtaining location information include GPS, Wi-Fi, beacons, and communication base stations. Because it is measured using IoT equipment, it already contains errors. To maintain a prespecified degree of privacy protection and enhance the usefulness of the data, we propose a method for reducing the total amount of added noise by considering errors already included in the measured values.

The principal contributions of this study are threefold. First, we introduce a novel concept of differential privacy tailored for location information, termed true-value-based *Geo-I* (*T-Geo-I*), which accounts for equipment noise. Second, we devise an anonymization algorithm that adheres to the *T-Geo-I* standard. Third, we demonstrate that the proposed *T-Geo-I* framework not only upholds the predefined privacy threshold but also reduces noise addition compared with existing methodologies.

The remainder of the paper is organized as follows: Section 2 reviews existing research related to differential privacy. Section 3 defines a new privacy metric and proposes a privacy protection algorithm that ensures compliance with this metric. Sections 4 and 5 detail the experimental method and the results, respectively. Section 6 discusses the experimental results of our proposed method. Finally, Section 7 concludes the study.

2. RELATED WORK

2.1. OVERVIEW OF LOCATION PRIVACY RESEARCH

A significant amount of research has been conducted on location information privacy [9–10]. One famous research field is differential privacy. *Geo-I* is famous for the differential privacy of location information [8].

According to recent research, *Geo-I* in indoor environments has been proposed [11]. The proposed framework introduces two distance calculation and received signal strength (RSS) generation methods based solely on RSS values as novel methods, which have been proven to perform.

Geo-I for task allocation in spatial crowdsourcing has been investigated [12]. An optimized global grouping with the adaptive local adjustment method OGAL with a convergence guarantee was proposed and proven that it works.

These methods do not consider measurement errors; therefore, our method can be applied to make them more efficient to enhance the usefulness of data.

Research on federated learning has been actively conducted recently [13]. Our method can also be incorporated into this. Details are explained in Section 2.10.

2.2. ϵ-DIFFERENTIAL PRIVACY

Differential privacy is extensively used as a strong mathematical definition to protect datasets without relying on attackers' prior information [14–16]. Rather than relying on encryption, differential privacy offers protection by adding noise to the data, and the results are calculated from the data. Because encryption is not involved, the computational cost of differential privacy is low, and it tends to be easy to introduce into many systems.

When mechanism K is a privacy protection function, $S \subseteq Range(K)$, $\epsilon \in R+$, and databases D and D' are adjacent, ϵ-differential privacy is satisfied when the following equation is satisfied. ϵ is a privacy level parameter and a positive number. When the privacy level parameter ϵ is large, the privacy level is low; when ϵ is small, i.e., close to 0, the degree of privacy protection is high. Adjacent means that the records are different in one place. For example, D represents a database with one record removed from D', otherwise D represents a database with one record of D' replaced by another record. This means that D and D' are adjacent.

$$Pr[K(D) \in S] \leq exp(\epsilon) \quad \times \quad Pr[K(D') \in S]. \tag{1}$$

This equation indicates that privacy is protected because different parts of the records cannot be identified if the results from adjacent databases are indistinguishable. For example, if an attacker knows all information except for a certain record A, it is possible to infer the data about A by back-calculating from the database result. Consequently, we can protect privacy by applying for protection according to this guarantee.

2.3. (E,Δ)-DIFFERENTIAL PRIVACY

Differential privacy is mathematically rigorous. It has been mathematically proven that a noise generation method based on the Laplace mechanism using the Laplace distribution has a probability density function ratio of less than the privacy level parameter ϵ in all ranges [17].

For example, for a noise generation method using a normal distribution noise, the ratio of probabilities becomes infinite at the tails of the distribution. Therefore, differential privacy is not guaranteed over the entire region.

However, it is too strict a definition to consider extreme points that seldom occur in reality. (ϵ,δ)-differential privacy allows cases wherein differential privacy is not satisfied if the probability is below a certain level [18].

When mechanism K is a privacy protection function, ϵ is a privacy level parameter, $S \subseteq$ Range(K), $\epsilon \in R+$, and databases D and D' are adjacent, if differential privacy based on the privacy level parameter is not satisfied with a probability less than or equal to δ, Equation 2 is satisfied.

$$Pr[K(D) \in S] \leq exp(\epsilon) \times Pr[K(D') \in S] + \delta. \quad (2)$$

2.4. LOCAL DIFFERENTIAL PRIVACY

The definition of ϵ-differential privacy refers to the protection of the database. Although this is guaranteed for databases that store data, it is not assumed that each data is sent to the server one by one each time. Therefore, the concept of local differential privacy has been proposed [19-20].

When x and x' represent databases of size 1 and protection is performed by mechanism A, for any output y, $\epsilon \in R+$, if Equation 3 is satisfied; for the privacy level parameter ϵ, it satisfies ϵ-local differential privacy.

$$Pr[A(x) = y] \leq exp(\epsilon) \times Pr[A(x') = y]. \quad (3)$$

This standard also allows you to protect your device before sending data to an untrusted server. Therefore, it is possible to collect and use data while protecting the data regardless of the trustworthiness of the server.

2.5. PLANAR LAPLACE MECHANISM

The planar Laplace mechanism is a typical privacy protection method based on differential privacy [17]. It uses the Laplace distribution to generate noise and adds it to the true value to protect privacy. When protecting individual data before sending it to the server, noise is added to each piece of data each time according to local differential privacy before sending it to the server. Because this method differs from encryption, it can protect user privacy with low computational costs. Therefore, it can be easily introduced into many systems. It can be executed on each user's IoT device or smartphone without a significant burden.

However, this method reduces the usefulness of the data. There is a trade-off between the usefulness of data and the degree of privacy protection. Many studies have been conducted to address this disadvantage, and our research is one of them to improve the usefulness of data.

2.6. ϵd_x-PRIVACY

Chatzikokolakis et al. [21] extended differential privacy, which is defined only in databases. $P(Z)$ denotes the probability distribution on Z. K:$X \rightarrow P(Z)$ denotes a mechanism in some domain X that provides a probability distribution in some domain Z. $Dx(x, x')$ is the hamming distance between x and x' on X. ϵ denotes a privacy level parameter, $\epsilon \in R+$, x, $x' \in X$, and $Z \subseteq Z$. If the mechanism K is expressed by Equation 4, ϵdx-privacy is guaranteed.

$$\frac{K(x)(Z)}{K(x')(Z)} \leq \epsilon d_X(x, x'). \quad (4)$$

This definition indicates that the more similar two databases are, the more similar the generated distributions should be.

2.7. GEO-INDISTINGUISHABILITY

Geo-I is a privacy guarantee standard for location information data. It has received particular attention among perturbation methods [22]. *Geo-I* applies ϵd_x-privacy to location information data. It also uses the concept of local differential privacy.

In Equation 5, X represents a set of points of interest, x,$x' \in X$, $d(x,x')$ denotes the distance between x and x' on the Euclidean plane, ϵ denotes a privacy level parameter, $\epsilon \in R+$, Z contains spatial points, and $Z \subseteq Z$. If the mechanism K is expressed by Equation 5, ϵ-*Geo-I* is guaranteed.

$$\frac{K(x)(Z)}{K(x')(Z)} \leq e^{\epsilon d(x,x')}. \quad (5)$$

2.8. PLANAR LAPLACE MECHANISM FOR *GEO-I*

The planar Laplace mechanism is used as a data protection method to satisfy *Geo-I* [22]. This is a method for position information wherein noise is generated from the privacy level parameter ϵ using the Laplace distribution and added to the true position.

For the noise radius value r, we substitute the noise calculated using Equation 6.

$$r_\epsilon(p) = -\frac{1}{\epsilon}\left(W_{-1}\left(\frac{p-1}{\epsilon}\right) + 1\right).$$

For the direction of noise value θ, we randomly calculate a value from the probability of a uniform distribution with [0, 2π). For p, we randomly calculate a value from the probability of a uniform distribution on [0,1), assign it to the true value x, and use $<r\cos\theta, r\sin\theta>$ as noise. We select the closest possible coordinate system to the coordinates with added noise and use that as the value after applying the mechanism. Function W_{-1} denotes Lambert's W function (the −1 branch). This operation guarantees ϵ-*Geo-I*. ϵ denotes a privacy level parameter, and $\epsilon \in R+$.

The probability of obscuring the true position x to x' is calculated using Equation 7. This planar Laplace mechanism rounds the decimal point of the position data. Equation 7 also considers the effect of rounding. d(x, x') denotes the distance between x and x' on the Euclidean plane. ϵ denotes a privacy level parameter.

$$D_{\epsilon}(x)(x') = \left(\frac{\epsilon^2}{2\pi}\right) e^{-\epsilon d(x,x')}. \tag{7}$$

2.9. TRUE-VALUE-BASED DIFFERENTIAL PRIVACY

Sei et al. [23] proposed the concept of true-value-based differential privacy (TDP). This is a privacy guarantee standard that considers the fact that the values measured using IoT devices contain errors.

The conventional method satisfies the specified degree of privacy protection for the measured value, i.e., "true value + measurement error." However, to meet these criteria, the privacy of the "true value" should be protected with a specified degree of privacy protection. Because noise in the form of measurement errors is already present, the amount of additional noise required to protect privacy is small for the necessary privacy parameter budget compared with the conventional method. Focusing on measurement errors, we attempt to reduce the total amount of noise added according to differential privacy.

For a database of size 1 for x and x', mechanism M is a function that adds error during measurement, and protection is provided by mechanism A. For any output y and $\epsilon \in R+$, when Equation 8 is satisfied, ϵ-differential privacy is satisfied. In addition, TDP assumes that the measurement error is based on a normal distribution.

$$Pr[A(M(x)) = y] \leq exp(\epsilon) \times Pr[A(M(x')) = y]. \tag{8}$$

Considering this concept, even if noise below an appropriate threshold is not added to the measured value, the prespecified degree of privacy protection can be maintained, and the total amount of added noise can be reduced.

TDP concentrates on one-dimensional data [23]. TDP aims to find the optimal maximum w that fulfills Equation 9.

$$e^{\epsilon} \geq \frac{\mathcal{V}(x + \Delta/2; \sigma^2, \Delta/\epsilon, w)}{\mathcal{V}(x - \Delta/2; \sigma^2, \Delta/\epsilon, w)} \tag{9}$$

where

$$\begin{aligned} &\mathrm{V}(\mathrm{x};\, \sigma^2, b, w) = \textstyle\int_{-\infty}^{\infty} \mathcal{N}(t; \sigma^2)\widehat{\mathcal{L}}(x - t; b, w)dt \\ &+ \mathcal{N}(x; \sigma^2) \textstyle\int_{-w}^{w} \mathcal{L}(t; b)dt \\ &= \frac{e^{-\frac{w+x}{b} - \frac{x^2}{2\sigma^2}}}{4b\sigma} \times \left\{\sigma e^{\frac{1}{2}\left(\frac{2bw+\sigma^2}{b^2} + \frac{x^2}{\sigma^2}\right)} \left[\mathrm{erfc}\left(\tfrac{b(w-x)+\sigma^2}{\sqrt{2}b\sigma}\right)\right.\right. \\ &\left.\left. + e^{\frac{2x}{b}} \mathrm{erfc}\left(\tfrac{b(w+x)+\sigma^2}{\sqrt{2}b\sigma}\right)\right] + 2\sqrt{\tfrac{2}{\pi}} b \left(e^{\frac{w}{b}} - 1\right) e^{\frac{x}{b}}\right\} \end{aligned}$$

and

$$\widehat{\mathcal{L}}(x; b, w) = \begin{cases} \int_{-w}^{w} \mathcal{L}(t; b)dt & x = 0 \\ \frac{e^{-x/b}}{2b} & x \geq w \\ \frac{e^{x/b}}{2b} & x \leq -w \\ 0 & otherwise. \end{cases}$$

Here, ϵ denotes a privacy level parameter, Δ means the range of possible values for numerical attitude, σ means the standard deviation of normal distribution, and b means the scale parameter of Laplace distribution (equal to Δ/ϵ).

The larger the threshold w, the more pronounced the reduction effects. TDP assumes that the measurement error adheres to a one-dimensional normal distribution $N(t;\sigma^2)$. If the measurement error diverges from a one-dimensional normal distribution, a fundamentally different mathematical discussion is required. Even with a one-dimensional normal distribution, as intricate as described by Equation 9, extending Equation 9 to two dimensions is not straightforward.

2.10. COMPOSITION THEOREM FOR HETEROGENEOUS MECHANISMS

Kairouz et al. [24] focused on privacy guarantees under k-fold composition. According to theorem 3.3 in [24], any k-fold adaptive composition of (ε, δ)-differentially private mechanisms satisfies the privacy guarantee. This means that the total privacy budget is obtained during composition.

2.11. FEDERATED LEARNING

Federated learning is a method that protects privacy by training machine learning models on each device [13, 25]. Each local device uses its data to train the model from the central server. Subsequently, only the extracted parameters are aggregated in the central server to improve the accuracy of the common model in the central server.

Federated learning of location information is also being researched [26–27]. For example, population modeling and population density can be estimated without the user having to send the true original data using the proposed method [27]. With federated learning, each device uses data to perform calculations and incorporates them into a machine learning model before sending the data to the server. It is highly compatible with local differential privacy. Our method is based on local differential privacy. There is a high possibility that our proposed method will be incorporated into federated learning to enhance the usefulness of data while maintaining a prespecified privacy protection level.

3. PROPOSED METHOD

3.1. TRUE-VALUE-BASED GEO-I (*T-GEO-I*)

We propose true-value-based geo-indistinguishability (*T-Geo-I*), a privacy protection standard for location information that considers measurement errors. This is a combination of *Geo-I*, which is a privacy protection standard related to location information, and TDP, which is a privacy protection standard that considers measurement errors.

TDP is focused on one-dimensional data. This leads to the meaningful proposition of amalgamating TDP with the privacy protection property of geo-indistinguishability for two-dimensional location information. The challenge in the theoretical analysis of the cumulative effect of measurement errors and differential privacy noise on two-dimensional location data is significant, rendering the direct application of the methodologies proposed in [23] unfeasible. In addition, the research on TDP, as discussed in [23], is confined to scenarios assuming a normal distribution of measurement errors. The uniqueness of the algorithm proposed in Section 3.2 of our study stems from its consideration of cases in which the measurement error does not conform to a normal distribution. This innovative approach significantly extends the applicability and relevance of TDP, particularly in contexts in which data distributions are non-normal. Obtained through simulation, our proposed algorithm is adaptable to any probability distribution. Typically, technologies for acquiring location data encompass GPS, Wi-Fi, beacons, and cellular base stations. Given the variety of devices and the indeterminate nature of measurement error distributions, the versatility of the proposed method in accommodating various error distributions is of substantial significance.

Let mechanism M be a function that adds error during measurement, X is a set of points of interest, $x, x' \in X$, $d(x, x')$ denotes the distance between x and x' on the Euclidean plane, ϵ denotes a privacy level parameter, $\epsilon \in R+$, Z contains spatial points, and $Z \subseteq Z$. ϵ-*T-Geo-I* is guaranteed when mechanism K satisfies Equation 10.

$$\frac{K(M(x))(Z)}{K(M(x'))(Z)} \leq e^{\epsilon d(x,x')}. \tag{10}$$

3.2. PRIVACY PROTECTION METHOD BASED ON *T-GEO-I*

Privacy protection method based on *T-Geo-I* is based on the planar Laplace mechanism. As mentioned in Section 2.4., because the planar Laplace mechanism has a low computational cost to protect privacy and is easy to use in various systems, our method incorporates this mechanism.

We propose a method wherein no noise is added to the data when the noise generated using the planar Laplace mechanism of *Geo-I* is below the threshold w; the noise is added to the data when the noise is the threshold w or above. Noise generation follows Section 2.7. The value generated using Equation 6 is the radius of the noise added to the measurement noise value, and the threshold w determines whether noise is added.

The problem with the proposed method is that it is difficult to solve the threshold value w analytically. In previous research [23], w was determined by calculation using mathematical formulas. We solve this problem by finding the threshold value w through simulation.

The pseudocode for the privacy protection method is shown in Algorithm 1. In the proposed method for analytically adding privacy noise, the noise radius r is calculated using Equation 11. For θ, we randomly calculate a value from the probability of a uniform distribution with $[0,2\pi)$. For p, we randomly calculate a value from the probability of a uniform distribution on [0,1). ϵ can be any positive value determined as a privacy level parameter.

Because of the proposed method, it is necessary to find an appropriate threshold value w for the noise radius r. The optimal threshold w value is the minimum value within the range that satisfies Equation 10.

Algorithm 2 illustrates the algorithm for determining the optimal threshold w. To confirm that Equation 10 is satisfied, a total noise probability density function is derived by combining the measurement error and privacy noise. Because the probability density function cannot be derived through calculation, it is derived by randomly generating ns samples as an experiment. A probability density function shifted by Δ is also derived. Differential privacy is satisfied when the ratio of the two probability density functions satisfies Equation 10. Because the accuracy of the probability density function is low in areas with few samples, only the areas with (1-δ) samples are checked. If differential privacy is satisfied, even with a sufficiently large threshold w, let w be infinite.

MeasurementNoise(), in the 8th line in Algorithm 2., returns the value obtained from the distribution of measurement errors. The distribution of measurement error is not limited to a normal distribution. The noise distribution may be any distribution and can be changed depending on the measuring equipment.

PrivacyNoise() in the 10th line in Algorithm 2. is the algorithm shown in Algorithm 1.

$$r_\epsilon(p) = -\frac{1}{\epsilon}\left(W_{-1}\left(\frac{p-1}{\epsilon}\right) + 1\right), \tag{11}$$

$$|PrivacyNoise(\epsilon, w)| = \begin{cases} r_\epsilon(p) \ (w < r_\epsilon(p)) \\ 0 \ (r_\epsilon(p) < w \ ). \end{cases} \tag{12}$$

Algorithm 1. Privacy protection mechanism for location information considering measurement errors.

```
Input: $\epsilon$ (Privacy level parameter), $v_x$, $v_y$ (Measured location values), $w$ (Threshold value)
Output: TDP value
Generate a random value $p$ from a uniform distribution [0,1)
$r \leftarrow r\epsilon(p)$
Generate a random value θ from a uniform distribution [0, 2π)
if $r < w$ then
    return $(v_x, v_y)$.
else
    return $(v_X + rcos\theta, v_y + rsin\theta)$.
end if
```

Algorithm 2. Algorithm for determining threshold w.

```
Input: $\epsilon$ (Privacy level parameter), $c$ (Width of a histogram), $\Delta$ (Distance of $x$ and $x'$), $\delta$ (Scope of verifying differential privacy), $\alpha$ (Multiple of $w$ to verify), $ns$ (Number of samples)
Output: Threshold $w$ used in the proposed method
1: for $w = \alpha, 2\alpha,...$ do
2:     isDF ← true
3: {Prepare two array variables as Histogram}
4:     $B \leftarrow b1 , b2 , . . .$
5:     $B' \leftarrow b1' , b2', ...$
6:     for $i$ = 1,. . .,$ns$ do
7: {Add measurement error}
8:     $v$ ←MeasurementNoise()
9: {Add Laplace noise considering threshold $w$}
10:    $v \leftarrow v + PrivacyNoise(\epsilon , w)$
11: {Calculate the corresponding bin of the histogram of value $v$.}
12:    $index \leftarrow \lceil |v|/c \rceil$
13:    $b_{\text{index}} \leftarrow b_{\text{index}} + 1$
14: {Calculate the corresponding bin of the histogram of value $|v| + \Delta$.}
15:    $index' \leftarrow \lceil (|v| + \Delta)/c \rceil$
16:    $b_{\text{index}} \leftarrow b_{\text{index}} + 1$
17:    end for
18: {Determine the scope to verify differential privacy}
19:    $sum \leftarrow 0$
20:    $threshold \leftarrow 0$
21:    for i = 1,. . . B'.length do
22:        $sum \leftarrow sum + b_I'$
23:        if $sum/ns > 1 - \delta$ then
24:            $threshold \leftarrow i$
25:            break
26:        end if
27:    end for
{Verify whether differential privacy is satisfied}
28:    for $i$ = 1,. . . ,threshold do
29:        if $b_i/b_i' >$ exp($\epsilon\Delta$) or $b_i'/b_i >$ exp($\epsilon\Delta$) then
30:            $isDF \leftarrow false$
31:            break
32:        end if
33:    end for
{Return value if differential privacy is not satisfied}
34:    if not isDF then
35:        return $w - \alpha$
36:    end if
37: end for
```

4. EXPERIMENT METHOD

4.1. SIMULATION METHOD

We simulated the proposed method. We compared the proposed method *T-Geo-I* with the planar Laplace mechanism for methods based on *Geo-I* [22] and TDP [23].

The simulation was performed in two scenarios. One involved performing experiments by setting a person's position to (0, 0) and adding noise as a numerical simulation. The other involved dividing people into grids and conducting a simulation experiment to count the number of people on each grid.

In the grid experiment, we used data generated using the Siafu simulation tool [28]. The Siafu tool is open-source software for obtaining data on human behavior using a typical human behavior model on a map. The setup includes 10,000 users interacting in a space measuring 8.4 km x 8.4 km, which includes businesses, restaurants, and parks. We used the data for this simulation based on previous research by Sei et al. [29].

In this experiment, the measurement error assumes 2 types, a normal distribution and a lognormal distribution. MeasurementNoise(), in the 8th line in Algorithm 2., returns noise based on a normal distribution or a lognormal distribution. Many studies on location information are based on the fact that GPS location measurement errors follow a normal distribution [30-33]. This study [34] showed the distributions that describe navigation positioning system errors more accurately include lognormal distributions. Therefore, the experiments were conducted by assuming that the measurement errors were based on a normal distribution and a lognormal distribution.

In the case wherein the measurement error is the lognormal distribution, experiments are only compared to *Geo-I*. As TDP is based on the case where the measurement error is the normal distribution, evaluations using TDP cannot be performed for the lognormal distribution.

As mentioned in the proposed method, the final noise is a combination of measurement errors and noise due to the Laplace mechanism. In the simulation, as shown in Fig. 1, the noise vector of measurement error due to the normal distribution and the noise vector due to the Laplace mechanism to satisfy differential privacy were added and used as the total noise.

The noise average and mean square error (MSE) are summarized in the results. Errors include both noise from the Laplace distribution for differential privacy and noise from the normal distribution as measurement errors.

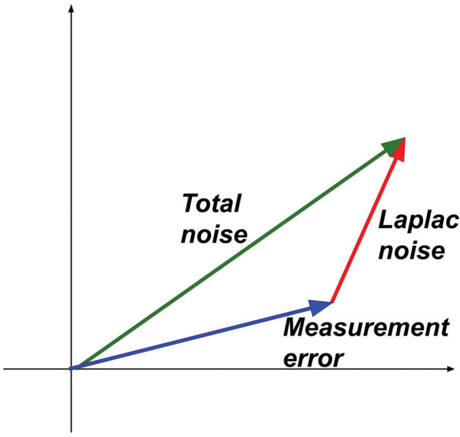


Fig. 1. Composition of angles

4.2. SIMULATION PARAMETERS

To find threshold w, the distance between x and x' Δ as $d(x, x')$, we conducted experiments with 1.0. The width of a histogram c is 0.5. Multiple of w to verify α is 0.5. The scope of verifying differential privacy δ is 10^{-3}. This means that we guarantee (ε, 10^{-3})-differential privacy. The number of samples ns is 10^8.

The measurement error was calculated from 2 types. One is a two-dimensional normal distribution with a standard deviation of 1.0. The other is the error by the radius from a lognormal distribution with a standard deviation of 1.0 and the angle is from a uniform distribution $[0, 2\pi)$. It is also used by MeasurementNoise() in Algorithm 2. For the noise generated from the Laplace distribution, we conducted experiments with $\epsilon = 1, 2, 5$, and 10.

In the numerical simulation, the number of samples is 10^8. In the Siafu simulation, the number of samples is 10^4. The space was divided into 500 × 500 squares, totaling 2,500 squares, and the noise average and noise MSE were calculated.

4.3. SIMULATION METHOD FOR TDP

TDP is focused on one-dimensional data basically [23]. In the method based on TDP, we consider x and y to be two independent variables.

According to Section 2.9., we generated noise with half the value ϵ and added it to x and y. For example, by adding noise generated from the Laplace distribution with $\varepsilon = 0.5$ for x and $\varepsilon = 0.5$ for y, we achieved total privacy protection of $\varepsilon = 1.0$.

5. EXPERIMENT RESULTS

In the case wherein the measurement error is the normal distribution, the total noise average and MSE of the numerical simulation are summarized in Tables 1 and 2, respectively. In the case wherein the measurement error is the normal distribution, the total noise average and MSE of the Siafu simulation are summarized in Tables 3 and 4, respectively. In the case wherein the measurement error is the normal distribution, the total noise average and MSE of the numerical simulation are summarized in Tables 5 and 6, respectively.

The total noise contains both Laplace noise for differential privacy and noise from the normal distribution as measurement errors. The results of the average amount of noise added to achieve differential privacy are summarized in Figs. 2, 3, and 4. When ε is close to 0, the noise is large.

According to all results, the proposed method has the smallest noise average and MSE compared with the other methods.

In the case wherein the measurement error is the normal distribution, the proposed method based on *T-Geo-I* reduced the noise average by up to 18% and 41% compared with methods based on *Geo-I* and TDP with numerical simulation, respectively. The proposed method based on *T-Geo-I* reduced the noise average by up to 15% and 36% compared with methods based on *Geo-I* and TDP with the Siafu simulation, respectively. The proposed method based on *T-Geo-I* reduced the noise MSE by up to 31% and 63% compared with *Geo-I* and TDP with numerical simulation, respectively. The proposed method based on *T-Geo-I* reduced the noise MSE by up to 17% and 38% compared with methods based on *Geo-I* and TDP with the Siafu simulation, respectively. The maximum reduction rate was achieved when $\varepsilon = 1, 2$.

In the case wherein the measurement error distribution is the lognormal distribution, the proposed *T-Geo-I* reduced the noise average and MSE by up to 60% and 67%, respectively, compared with *Geo-I* with numerical simulation. The maximum reduction rate was achieved when $\varepsilon = 1$.

In the case of $\epsilon = 5$ and 10, the result indicates that differential privacy is satisfied with only the measurement error without any noise addition because of the Laplace distribution. When $\epsilon = 10$, the noise averages of methods based on *T-Geo-I* and TDP are almost the same. This indicates that both methods do not add nearly any Laplace noise because differential privacy is almost satisfied with only the standard deviation when $\epsilon = 10$.

The proposed method can reduce the average amount of noise and is expected to enhance the usefulness of the data.

We tested them on a MacBook Air (M1, 2020), an Apple M1 CPU, and 16 GB of memory using Python. It takes

5 h to generate 10^8 Laplace noises. It takes 5 min to read the data of 10^8 Laplace noises already generated.

After the data are read, it takes 5 min for each value of w to create a histogram and verify whether differential privacy is satisfied.

We also experimented to see how much time it takes to protect privacy in a real environment. We measured the calculation time for acquiring location information and adding noise to the location information using an iPhone 13 mini. The average time value was calculated by measuring 100 times. The result is Fig. 5. The computation time for all methods was almost the same. Privacy protection can be achieved in a short time of 270-290 ms. This means that the proposed method is not algorithmically inefficient.

Table 1. Comparison of total noise average with numerical simulation (measurement error of normal distribution)

ϵ	*w* for *T-Geo-I*	*T-Geo-I* (noise average)	*Geo-I* (noise average)	TDP (noise average)
1	2.5	2.02	2.41	3.46
2	2.5	1.33	1.64	1.92
5	inf	1.25	1.33	1.27
10	inf	1.25	1.27	1.25

Table 2. Comparison of total noise MSE with numerical simulation (measurement error of normal distribution)

ϵ	*w* for *T-Geo-I*	*T-Geo-I* (noise MSE)	*Geo-I* (noise MSE)	TDP (noise MSE)
1	2.5	6.54	7.99	17.72
2	2.5	2.39	3.50	5.31
5	inf	1.99	2.23	2.07
10	inf	1.99	2.05	2.00

Table 3. Comparison of total noise average with Siafu simulation (measurement error of normal distribution).

ϵ	*w* for *T-Geo-I*	*T-Geo-I* (noise average)	*Geo-I* (noise average)	TDP (noise average)
1	2.5	1.96	2.28	3.09
2	2.5	1.40	1.65	1.84
5	inf	1.36	1.40	1.36
10	inf	1.36	1.37	1.36

Table 4. Comparison of total noise MSE with Siafu simulation (measurement error of normal distribution).

ϵ	*w* for *T-Geo-I*	*T-Geo-I* (noise MSE)	*Geo-I* (noise MSE)	TDP (noise MSE)
1	2.5	1.96	2.28	3.09
2	2.5	1.40	1.65	1.84
5	inf	1.36	1.40	1.36
10	inf	1.36	1.37	1.36

Table 5. Comparison of total noise average with numerical simulation (measurement error of lognormal distribution).

ϵ	*w* for *T-Geo-I*	*T-Geo-I* (noise average)	*Geo-I* (noise average)
1	inf	1.65	4.17
2	inf	1.65	3.74
5	inf	1.65	3.60
10	inf	1.65	3.58

Table 6. Comparison of total noise MSE with numerical simulation (measurement error of lognormal distribution).

ϵ	*w* for *T-Geo-I*	*T-Geo-I* (noise MSE)	*Geo-I* (noise MSE)
1	inf	7.39	22.85
2	inf	7.39	18.35
5	inf	7.39	17.09
10	inf	7.39	16.91

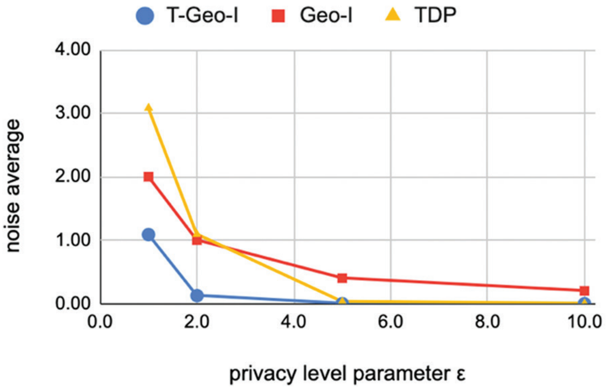


Fig. 2. Average amount of noise added to achieve differential privacy with numerical simulation (measurement error of normal distribution)

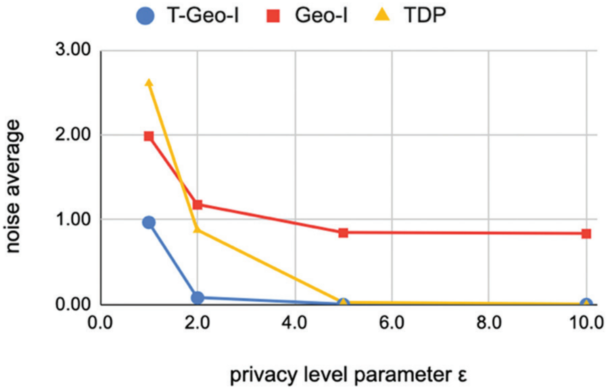


Fig. 3. Average amount of noise added to achieve differential privacy with Siafu simulation (measurement error of normal distribution)

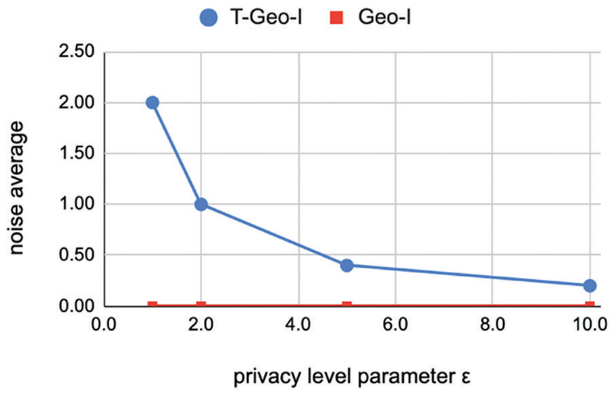


Fig. 4. Average amount of noise added to achieve differential privacy with numerical simulation (measurement error of lognormal distribution)

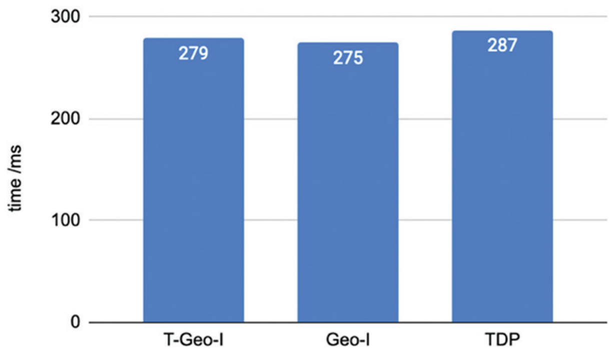


Fig. 5. The time required to measure the location information on the user's smartphone device and to apply differential privacy noise

6. DISCUSSION

In Apple's development, the privacy level parameter ϵ is equal to 1 or 2 per datum [35]. For example, Apple's differential privacy team used ϵ = 2, 4, and 8 for their experiment evaluations [36]. In the study that proposed RAPPOR by Google, $\epsilon = \log(3)$ was used as the main parameter [37]. In TDP study [23], ϵ is set in the range 1–10. Therefore, we experimented with ϵ = 1, 2, 5, and 10.

It was confirmed that the noise average was reduced not only in the numerical simulation but also in the Siafu simulation. The Siafu simulation is based on a typical human behavior model on a map. This means that we can expect to enhance the usefulness of data even in real-life situations.

As shown in Figs. 2, 3, and 4, the noise average was higher at a higher degree of privacy protection. This means that the effect of noise reduction using the proposed method is high if the degree of privacy protection is high. According to the results in Tables 1–6, the greatest reduction effect is obtained when ε = 1 and 2. Because the Laplace noise is small when ε = 5 and 10, the reduction in the total noise of the measurement error and the Laplace noise is small.

The case where *w=inf* means that no Laplace noise is added. When *w=inf*, the noise regarding the proposed *T-Geo-I* is from only measurement error. In other words, $(\varepsilon, 10^{-3})$-differential privacy is satisfied even without adding any Laplace noise. It is shown that there are cases wherein privacy can be protected using only measurement errors. Note that the proposed method satisfies differential privacy at the specified level. In other words, the existing methods add unnecessary noise beyond the specified level.

It takes more than 5 h to calculate the threshold w when the number of samples is 10^8. However, once the value of *w* is calculated, the determined value w can be repeatedly used for actual privacy protection. 10^8 Laplace noise generation is necessary for the simulation to determine the threshold *w* and only needs to be done once on the server side.

On the contrary, actual privacy protection takes a very short time. As shown in the newly added Fig. 5, actual privacy protection has a low computational cost. The computation time for all methods was almost the same. Privacy protection can be achieved in a short time of 270-290 ms. This shows that the proposed method is not algorithmically inefficient. Because this method has a very low computational cost, it can be easily introduced into various systems. The usefulness of the data can be improved compared with conventional methods.

The conventional method TDP assumes a normal distribution of measurement errors [23]. Our method is not limited to normal distributions. An appropriate threshold value w can be determined by simulation of any distribution. This is an advantage of our method.

In this experiment, we assumed a normal distribution and a lognormal distribution for measurement errors. Many studies have been conducted on measurement errors in location information. They are affected by various factors such as radio waves and weather conditions. They cannot be determined in one way. There is also research on simulation measurement errors [34, 38]. In the future, experiments are expected to be conducted on measurement errors in various situations.

The disadvantage is that the simulation for finding the threshold value *w* is computationally expensive. In the future, methods for determining the threshold value *w* based on the proof of mathematical formulas instead of simulation are expected.

Our method does not consider continuous location information. By acquiring continuous location information based on the trajectory of a person's movement, the risk of estimating the person's true location is increased [39–40]. In the future, we intend to address these issues.

7. CONCLUSION

Systems that collect location information and publish statistics, such as those that publish congestion information, have been extensively employed. These systems use differential privacy to ensure the privacy of user data. Privacy protection using the Laplace mechanism based on differential privacy adds noise, which reduces the usefulness of the data when the degree of privacy protection is high. Therefore, we focus on the fact that the values obtained by measurement devices contain errors and propose a location information privacy protection method that reduces the amount of added noise.

In the case wherein the measurement error is the normal distribution, the proposed method based on *T-Geo-I* succeeded in reducing the noise average by up to 18% and 41% compared with methods based on *Geo-I* and TDP, respectively, while maintaining a prespecified level of privacy in 10^8 samples of numerical data. It also reduced the noise MSE by up to 31% and 63% compared with methods based on *Geo-I* and TDP, respectively. The proposed method based on *T-Geo-I* reduced the noise average by up to 15% and 36% compared with methods based on *Geo-I* and TDP, respectively, in a location simulation of the human behavior of 10^4 users on a map using a typical human behavior model. It also reduced the noise MSE by up to 17% and 38% compared with methods based on *Geo-I* and TDP, respectively.

In the case wherein the measurement error is the log-normal distribution, the proposed method based on *T-Geo-I* succeeded in reducing the noise average and MSE by up to 60% and 67%, respectively, compared with methods based on *Geo-I*, while maintaining a prespecified level of privacy in 10^8 samples of numerical data.

The maximum reduction rate was achieved when ε is small: the privacy protection level high.

These findings demonstrate that our method can improve the usefulness of data while maintaining a prespecified privacy protection level.

8. REFERENCE

[1] T. Alam, B. Rababah, A. Ali, S. Qamar, "Distributed Intelligence at the Edge on IoT Networks", Annals of Emerging Technologies in Computing, Vol. 4, No. 5, 2020, pp. 1-18.

[2] G. Muneeswari, A. Ahilan, R. Rajeshwari, K. Kannan, C. J. C. Singh, "Trust and Energy-Aware Routing Protocol for Wireless Sensor Networks Based on Secure Routing", International Journal of Electrical and Computer Engineering Systems, Vol. 14, No. 9, 2023, pp. 1015-1022.

[3] D. Liu, X. Gao, H. Wang, "Location privacy breach: Apps are watching you in background", Proceedings of the IEEE 37th International Conference on Distributed Computing Systems, Atlanta, GA, USA, 5-8 June 2017, pp. 2423-2429.

[4] S. Kumar et al. "Protecting location privacy in cloud services", Journal of Discrete Mathematical Sciences and Cryptography, Vol. 25, No. 4, 2022, pp. 1053-1062.

[5] K. S. Saraswathy, S. S. Sujatha, "Using Attribute-Based Access Control, Efficient Data Access in the Cloud with Authorized Search" International Journal of Electrical and Computer Engineering Systems, Vol. 13, No. 7, 2022, pp. 569-575.

[6] G. Sun et al. "Location Privacy Preservation for Mobile Users in Location-Based Services", IEEE Access, Vol. 7, 2019, pp. 87425-87438.

[7] N. Ahmed, Z. Deng, I. Memon, F. Hassan, K. H. Mohammadani, R. Iqbal, "A Survey on Location Privacy Attacks and Prevention Deployed with IoT in Vehicular Networks", Wireless Communications and Mobile Computing, Vol. 2022, 2022.

[8] E. P. de Mattos, A. C. S. A. Domingues, B. P. Santos, H. S. Ramos, A. A. F. Loureiro, "The Impact of Mobility on Location Privacy: A Perspective on Smart Mobility", IEEE Systems Journal, Vol. 16, No. 4, 2022 pp. 5509-5520.

[9] S. Özdal Oktay, S. Heitmann, C. Kray, "Linking location privacy, digital sovereignty and location-based services: a meta review", Journal of Location Based Services, Vol. 18, No. 1, 2024, pp. 1-52.

[10] M. K. Gupta, A. K. Rai, B. Pandey, A. Gupta, V. K. Verma, "Big Data Privacy: A Survey Paper", Proceedings of the International Conference on IoT, Communication and Automation Technology, Gorakhpur, India, 2023, pp. 1-6.

[11] A. Fathalizadeh, V. Moghtadaiee, M. Alishahi, "Indoor Geo-Indistinguishability: Adopting Differential Privacy for Indoor Location Data Protection", IEEE Transactions on Emerging Topics in Computing, 2023. (in press)

[12] P. Zhang, X. Cheng, S. Su, N. Wang, "Task Allocation Under Geo-Indistinguishability via Group-Based Noise Addition", IEEE Transactions on Big Data, Vol. 9, No. 3, 2023, pp. 860-877.

[13] E. T. Martínez Beltrán et al. "Decentralized Federated Learning: Fundamentals, State of the Art, Frameworks, Trends, and Challenges", Proceedings of the IEEE Communications Surveys & Tutorials, Vol. 25, No. 4, 2023, pp. 2983-3013.

[14] C. Dwork, "Differential Privacy", Proceedings of the 33rd International Colloquium on Automata, Languages, and Programming, Vencie, Italy, 10-14 July 2006, pp. 1-12.

[15] C. Dwork, "Differential Privacy: A Survey of Results", Proceedings of the International Conference on Theory and Applications of Models of Computation, Xi'an, China, 25-29 April 2008, pp. 1-19.

[16] C. Dwork, A. Roth, "The algorithmic foundations of differential privacy", Foundations and Trends in Theoretical Computer Science, Vol. 9, No. 3-4, 2014, pp. 211-407.

[17] C. Dwork, F. McSherry, K. Nissim, A. Smith, "Calibrating noise to sensitivity in private data analysis", Proceedings of the Theory of Cryptography: Third Theory of Cryptography Conference, New York, NY, USA, 4-7 March 2006, pp. 265-284.

[18] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, M. Naor, "Our data, ourselves: Privacy via distributed noise generation", Advances in Cryptology – EUROCRYPT 2006, Proceedings of the 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, 28 May - 1 June 2006, pp. 486-503.

[19] R. Dewri, "Local differential perturbations: Location privacy under approximate knowledge attackers", IEEE Transactions on Mobile Computing, Vol. 12, No. 12, 2012, pp. 2360-2372.

[20] J. C. Duchi, M. I. Jordan, M. J. Wainwright, "Local privacy and statistical minimax rates", Proceedings of the IEEE 54th Annual Symposium on Foundations of Computer Science, Berkeley, CA, USA, 26-29 October 2013, pp. 429-438.

[21] K. Chatzikokolakis, M. E. Andrés, N. E. Bordenabe, C. Palamidessi, "Broadening the Scope of Differential Privacy Using Metrics", Proceedings of Privacy Enhancing Technologies: 13th International Symposium, Bloomington, IN, USA, 10-12 July 2013, pp. 82-102.

[22] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems", Proceedings of the ACM SIGSAC Conference on Computer & Communications Security, November 2013, pp. 901-914.

[23] Y. Sei, A. Ohsuga, "Private true data mining: Differential privacy featuring errors to manage Internet-of-Things data", IEEE Access, Vol. 10, 2022, pp. 8738-8757.

[24] P. Kairouz, S. Oh, P. Viswanath, "The composition theorem for differential privacy", Proceedings of the 23rd International Conference on Machine Learning, Lille, France, 2015, pp. 1376-1385.

[25] J. Konečný, H. B. McMahan, D. Ramage, P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence", arXiv:1610.02527, 2016.

[26] Y. Zhang, Y. Lu, F. Liu, "A systematic survey for differential privacy techniques in federated learning", Journal of Information Security, Vol. 14, No. 2, 2023, pp. 111-135.

[27] Z. Zong, M. Yang, J. Ley, A. Markopoulou, C. Butts, "Privacy by Projection: Federated Population Density Estimation by Projecting on Random Features", Proceedings on Privacy Enhancing Technologies, Vol. 2023, No. 1, 2023, pp. 309-324.

[28] M. Martin, P. Nurmi, "A Generic Large Scale Simulator for Ubiquitous Computing", Proceedings of the Third Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services, San Jose, CA, USA 17-21 July 2006, pp. 1-3.

[29] Y. Sei, A. Ohsuga, "Location Anonymization With Considering Errors and Existence Probability", IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 47, No. 12, 2016, pp. 3207-3218.

[30] P. Chao, W. Hua, R. Mao, J. Xu, X. Zhou, "A Survey and Quantitative Study on Map Inference Algorithms From GPS Trajectories", IEEE Transactions on Knowledge and Data Engineering, Vol. 34, No. 1, 2020, pp. 15-28.

[31] E. Frentzos, K. Gratsias, Y. Theodoridis, "On the Effect of Location Uncertainty in Spatial Querying", IEEE transactions on Knowledge and Data Engineering, Vol. 21, No. 3, 2008, pp. 366-383.

[32] D. Zhang, Z. Chang, S. Wu, Y. Yuan, K.-L. Tan, G. Chen, "Continuous Trajectory Similarity Search for Online Outlier Detection", IEEE Transactions on Knowledge and Data Engineering, Vol. 34, No. 10, 2020, pp. 4690-4704.

[33] T. Ogino, "GPS Improvement System Using Short-Range Communication", Proceedings of the International Conference on Computing, Networking and Communications, Maui, HI, USA, 5-8 March 2018, pp. 82-87.

[34] M. Specht, "Consistency of the Empirical Distributions of Navigation Positioning System Errors with Theoretical Distributions—Comparative Analysis of the DGPS and EGNOS Systems in the Years 2006 and 2014", Sensors, Vol. 21, No. 1, 2020, p. 31.

[35] J. Tang, A. Korolova, X. Bai, X. Wang, X. Wang, "Privacy Loss in Apple's Implementation of Differential Privacy on MacOS 10.12", arXiv:1709.02753, 2017

[36] Differential Privacy Team, "Learning with privacy at scale", Apple Machine Learning Research, Apple, December 2017.

[37] Ú. Erlingsson, V. Pihur, A. Korolova, "RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response", Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, November 2014, pp. 1054-1067.

[38] A. El Abbous, N. Samanta, "A modeling of GPS error distributions", Proceedings of the European Navigation Conference, Lausanne, Switzerland, 9-12 May 2017. pp. 119-127.

[39] Y. Zhao, J. Chen, "Vector-Indistinguishability: Location Dependency Based Privacy Protection for Successive Location Data", IEEE Transactions on Computers, 2023. (in press)

[40] X. Sun et al. "Synthesizing Realistic Trajectory Data with Differential Privacy", IEEE Transactions on Intelligent Transportation Systems, Vol. 24, No. 5, 2023, pp. 5502-5515.

An empirical study on differences between self-assessed and measured real risk in online behaviour

Original Scientific Paper

Krešimir Šolić
J. J. Strossmayer University of Osijek,
Faculty of Medicine, Department of Medical Statistics and Medical Informatics
Josipa Huttlera 4, Osijek, Croatia
kresimir@mefos.hr

Robert Idlbek
J. J. Strossmayer University of Osijek,
Faculty of Tourism and Rural Development, Department of Computer Science
Vukovarska 17, Požega, Croatia
ridlbek@ftrr.hr

Tena Velki
J. J. Strossmayer University of Osijek,
Faculty of Education, Department of Social Sciences
Ul. Cara Hadrijana 10, Osijek, Croatia
tvelki@foozos.hr

Abstract *– As the leading cause of security breaches is human susceptibility to hackers' deception, the riskiness of an individual's online behaviour and low awareness regarding privacy protection significantly influence the overall security of an information system. Thus, this study aimed to compare self-assessed and measured real risk in online behaviour among online users. The additional aim was to modify the questionnaire by replacing the existing trick question about password quality with the new questions on accepting the terms and conditions. An international online Behavioral Cognitive Internet Security Questionnaire (BCISQ), validated in previous studies, was used for data collection. The examinees involved in this study were 278 students from different faculties. The results showed a relatively high level of risk in online behaviour, as 22.7% of examinees revealed their passwords.*

In comparison, only 10.8% read the consent statement. Students who behave in a riskier manner self-assess themselves as being significantly safer in online behaviour, which is contradictory. They also performed worse in all other examined variables. The new version of the simulation subscale, with improved internal consistency and reliability (Cronbach's Alfa=0.810), consists of only three items, which are questions used in the previous version, without adding any of the two tested trick questions. Generally, this study concludes that, on average, information security awareness is still low among online users and that even the ones realistically acting riskier believe they are acting more safely.

Keywords: *information security, information system, security awareness, user behaviour*

Received: November 12, 2023; Received in revised form: January 21, 2024; Accepted: January 22, 2024

1. INTRODUCTION

The direct or indirect aim of the security breach on an information system is basically to gain some financial benefit. Therefore, in the beginning, the information systems of the banking sector were best protected by additional national and international regulations. Onwards, security experts, who were primarily managers of security regulations, were focused on information security policies in business companies and healthcare information systems, as loss of public reputation can indirectly cause financial loss. Nowadays, information security and privacy protection focus on any information system in the business and non-profit sectors and public and private areas. However, for many years, the information system user has been identified as the weakest link in information security protocols, as the leading cause of security breaches is human susceptibility to hacker's deception [1]. So, the human factor still represents the central junction regarding cyber-attacks [2]. Therefore, influencing user behaviour, raising security awareness, and protecting an individual's privacy will increase the overall security of an information system.

Level of knowledge, behaviour toward following security guidelines and learning inertia can significantly influence information security awareness [3]. However, users are susceptible to social engineering despite targeted education [4]. Furthermore, even highly aware online users often give personal data away voluntarily and behave in a high-risk manner on the internet [5]. It is very worrying and confusing, but no comprehensive explanation for this privacy paradox has been found so far [6]. However, although insufficient, education still significantly impacts increasing safety in online behaviour [7].

Further cyber security training to improve digital trust is needed to raise individuals' awareness [8-11]. New concepts to solve the problem of risky behaviour and low-security awareness should combine periodic education regularly with some notification system. Some studies also suggest that future learning models should use more interactive educational methods and should be based on simulation procedures [12, 13].

Online user's text-based passwords are still the first line of defence. However, they are still weak in securing all kinds of information systems. Users' careless security behaviour, involving password reuse, writing down and sharing passwords, and creating short or low-quality passwords are the main problems related to password security issues [14]. Modelling users' risky online behaviour based on analysing millions of passwords, both the most frequent passwords and how users create new passwords, can be helpful to hackers [15, 16].

The quality of the password, e.g., how the password is constructed, differs between students, average users, and professionals [17]. Average online users like having and using usernames and passwords with similar characters - the first few digits or the last few digits in a decade system, while the most used unique character is the underscore sign [18]. Additionally, male users have significantly stronger passwords than female ones, and password complexity decreases with age [19]. Also, 72% of users based their passwords on a single word or used a simple sequence of digits. Meanwhile, 39% of examined passwords were found in word lists of previous password leaks [19]. An additional paradox regarding the quality of passwords is as follows: a simple one is easier to remember, but a complicated one is more secure from being guessed [20].

Most research studies regarding passwords are focused on their quality. However, as the most essential property of a password is its secrecy, other properties such as length and the combination of special characters are becoming irrelevant. Findings in previous studies have shown that up to three out of four average online users will, in some cases, reveal their passwords, mainly to a friend, college, or authority figure. The easiest way to find someone's password is to ask for it. However, over the last few years, a promising trend has shown specific improvements [21].

Password disclosure becomes a big problem when someone logs in and thus impersonates the system during identification. That is why advanced additional confirmation methods, such as biometrics and blockchain technology, are increasingly used during authentication [22-24]. The most secure way is to use the three-factor authentication (3FA) scheme to identify itself through three categories of authentication factors (knowledge, possession and inherence): something you know, have, and are.

Many online users, or even most, have never read the terms and conditions but accept them without reading and understanding. A probable reason is that terms and conditions are verbose and contain legal jargon [25]. Accepting something online without reading it can lead to significant information security and privacy risks, and younger online users are more careless regarding reading terms and conditions [26]. Reading terms and conditions is related to concern for privacy, positive perceptions about notice comprehension, and higher trust in the notice. Three-quarters of participants included in one study skipped reading privacy policies, as they view policies as a nuisance and ignore them [27]. The results of another study have shown that most participants will skip reading the privacy policy if it is not presented by default [28].

This study aimed to analyse users' risky online behaviour to compare self-assessed and measured actual levels of risk. It also examined the awareness and knowledge of information security and privacy protection issues. The additional aim was to modify the questionnaire by replacing the existing trick question about password quality with the new trick question on accepting terms and conditions to improve the internal consistency and reliability of the simulation subscale. The study was based on a Croatian version of the previously developed and statistically validated international online questionnaire: the Behavioral Cognitive Internet Security Questionnaire (BCISQ) [29].

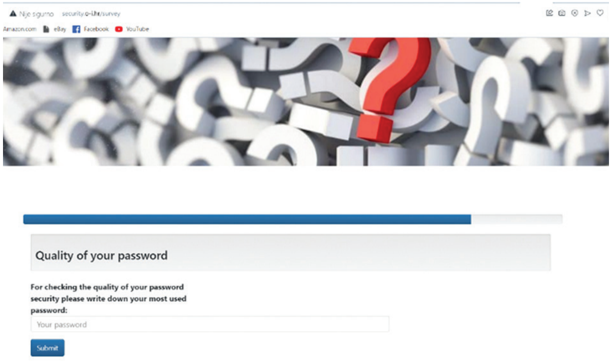
The BCISQ was chosen as it measures real risky online behaviour with its simulation subscale compared to similar solutions. Many empirical studies on this subject have been made. However, only several statistically validated questionnaires are developed as the basis for empirical studies dealing with information system users' risky behaviour. One of the most used is the SeBIS (Security Behavior Intentions Scale), which was developed in the USA and published in 2016 [30]. Then, in the same year, the FMS (Four Measurements Scales) was designed and validated in Turkey [31]. Then, the HAIS Q (Human Aspects of Information Security) was developed in Australia, with a validated version published in 2017 [32].

2. MATERIALS & METHODS

An internationally validated Behavioral Cognitive Internet Security Questionnaire (BCISQ) was used for data collection. This questionnaire has only an online version currently available in four languages at http://security.o-i.hr. The BCISQ consists of four subscales and measures: simulated risky online behaviour, self-assessed risk of online behaviour, cognitive awareness of online risks, and the importance of safe online usage. The questionnaire uses 17 items divided into subscales and has additional demographic questions [29]. In this research, a Croatian version of a questionnaire was used.

This study primarily focuses on measuring a real online risk by analysing the data gathered with the first subscale that simulates real online risky situations, emphasising the trick question about password quality and, with additional, new trick question examining how much online users read terms and conditions. However, all collected data are correlated with the other three subscales and demographic questions in further analysis.

The simulation subscale consists of four questions, with the first two asking if the examinee would like to receive notifications from third-party partners about similar studies and free antivirus software from third-party partners via email. The third question asks the examinee to leave an email address, and the fourth question, positioned at the end of the BCISQ questionnaire, is a trick question asking the examinee to reveal their most used password. A trick question is constructed so that the examinee is deceived by scientific and anonymous research to write down a password to help researchers examine the quality of the password's security (Fig. 1).

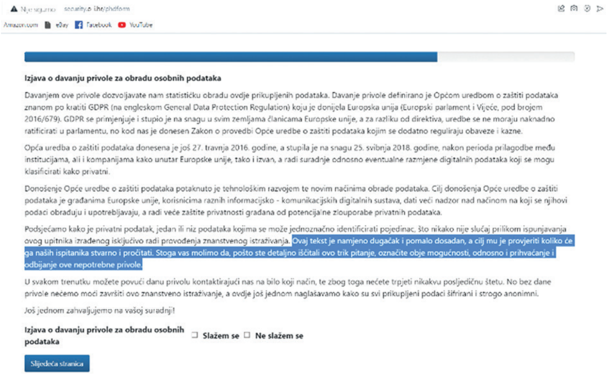


Quality of your password

For checking the quality of your password security please write down your most used password:

Fig. 1. Visual of the trick question regarding password quality

As participants in this study were Croatian students, an additional question was constructed for this research only in the Croatian version of the BCISQ questionnaire. This additional, new trick question was named Statement of Consent for processing personal data and has 318 words of text explaining what the GDPR is and why this research is essential. After approximately 80% of a text, there is an instruction for the examinee to mark both squares: to both agree and disagree (Fig. 2).



Izjava o davanju privole za obradu osobnih podataka

Davanjem ove privole dozvoljavate nam statističku obradu ovdje prikupljenih podataka. Davanje privole definirano je Općom uredbom o zaštiti podataka znanom po kratiti GDPR (na engleskom General Data Protection Regulation) koju je donijela Europska unija (Europski parlament i Vijeće, pod brojem 2016/679). GDPR se primjenjuje i stupio je na snagu u svim zemljama članicama Europske unije, a za razliku od direktiva, uredbe se ne moraju naknadno ratificirati u parlamentu, no kod nas je donesen Zakon o provedbi Opće uredbe o zaštiti podataka kojim se dodatno reguliraju obaveze i kazne.

Opća uredba o zaštiti podataka donesena je još 27. travnja 2016. godine, a stupila je na snagu 25. svibnja 2018. godine, nakon perioda prilagodbe među institucijama, ali i kompanijama kako unutar Europske unije, tako i izvan, a radi suradnje odnosno eventualne razmjene digitalnih podataka koji se mogu klasificirati kako privatni.

Donošenje Opće uredbe o zaštiti podataka potaknuto je tehnološkim razvojem te novim načinima obrade podataka. Cilj donošenja Opće uredbe o zaštiti podataka je građanima Europske unije, korisnicima raznih informacijsko - komunikacijskih digitalnih sustava, dati veći nadzor nad načinom na koji se njihovi podaci obrađuju i upotrebljavaju, a radi veće zaštite privatnosti građana od potencijalne zlouporabe privatnih podataka.

Podsjećamo kako je privatni podatak, jedan ili niz podataka kojima se može jednoznačno identificirati pojedinac, što nikako nije slučaj prilikom ispunjavanja ovog upitnika izrađenog isključivo radi provođenja znanstvenog istraživanja. Ovaj tekst je namjeno dugačak i pomalo dosadan, a cilj mu je provjeriti koliko će ga naših ispitanika stvarno i pročitati. Stoga vas molimo da, pošto ste detaljno iščitali ovo trik pitanje, označite obje mogućnosti, odnosno i prihvaćanje i odbijanje ove nepotrebne privole.

U svakom trenutku možete povući danu privolu kontaktirajući nas na bilo koji način, te zbog toga nećete trpjeti nikakvu posljedičnu štetu. No bez dane privole nećemo moći završiti ovo znanstveno istraživanje, a ovdje još jednom naglašavamo kako su svi prikupljeni podaci šifrirani i strogo anonimni.

Još jednom zahvaljujemo na vašoj suradnji!

Izjava o davanju privole za obradu osobnih podataka ☐ **Slažem se** ☐ **Ne slažem se**

Fig. 2. Position of explanation in the trick question regarding (not) reading terms and conditions

Here is the text of the new question, translated to English under the title: Statement of consent for processing personal data

By giving this consent, you allow us to statistically process the data collected here. Consent is defined by the General Data Protection Regulation, known by the abbreviation GDPR (in English General Data Protection Regulation), adopted by the European Union (European Parliament and Council, under number 2016/679). The GDPR is applied and enforced in all European Union member states. Unlike directives, regulations do not have to be subsequently ratified in parliament. However, we have adopted the Law on Implementing the General Data Protection Regulation, which regulates obligations and penalties.

The General Data Protection Regulation was adopted on April 27, 2016. It came into force on May 25, 2018, after a period of adjustment among institutions and companies both within and outside the European Union and for cooperation, i.e., the eventual exchange of digital data that can be classified as private.

The adoption of the General Data Protection Regulation was prompted by technological development and new ways of data processing. The goal of adopting the General Regulation on Data Protection is to give citizens of the European Union, users of various information and communication digital systems, greater control over how their data is processed and used and for more excellent protection of citizens' privacy from potential misuse of private data.

We remind you that private data is one or a series of data that can uniquely identify an individual, which is by no means the case when filling out this questionnaire created solely to conduct scientific research. This text is deliberately lengthy and somewhat dull, and its goal is to check how many of our respondents will read it. Therefore, after reading this trick question in detail, we instruct you to mark both options to accept and decline this unnecessary consent.

You can withdraw your consent at any time by contacting us in any way, and you will not suffer any consequential damages. However, without consent, we cannot complete this scientific research, and here, we emphasise that all collected data is encrypted and strictly anonymous.

Thank you once again for your cooperation!

□ *I agree* □ *I do not agree*

This new trick question has been planned to examine how many examinees read the Statement of consent. So, the text itself is of no importance. It is not very informative but deliberately long. At the same time, the crucial sentence with instructions for examinees is underlined only here in the translation text.

The authors attempted to replace the existing trick question asking for a password with the new trick question by examining the reading of the Statement of consent, as previous research had shown that the question on a password decreased the stability (internal consistency and reliability) of the simulation subscale of risky online behaviour [33]. It was also unclear if the revealed password was real and still actual for this particular examinee, as even the examinee could leave this field blank. Many of them wrote down something in a way that they did not want to write down their password. Before analysis, all passwords were inspected and removed if they did not look like actual passwords.

Standard statistical methods were applied to the collected data, where each used statistical model is pointed out at the bottom of each table. Categorical data are presented with absolute and relative frequencies. At the same time, the Chi-square Test was used to compare categorical data between independent groups. A normality test was applied to each distribution of numerical data to choose a parametric or nonparametric test and how to present the average value (median or arithmetic mean). Because distributions of examined numerical data did not follow Gaussian normal distribution, data were presented with median, interquartile and total range. They were tested with a nonparametric Mann-Whitney Test for independent samples. When modifying the examined subscale, Cronbach's alpha coefficient was calculated to estimate each version's internal consistency and reliability. Analysis was done in the statistical tool MedCalc (version 20.218, 64-bit, MedCalc Software Ltd), with statistical significance set at α=0.05, where all P values were two-tailed.

The examinees were 278 students from different J. J. Strossmayer University of Osijek faculties. There were 48 students from the Faculty of Education, including 28 studying rehabilitation; 73 from the Faculty of Medicine, including 33 studying to be laboratory technicians; 49 from the Faculty of Dental Medicine, including nursing and physiotherapy; 19 from the Faculty of Tourism and Rural Development, and 11 from the Faculty of Economics, while the rest were from other faculties and gathered mainly on the university campus. As future engineers have already proven, in a previous study, to be a specific sample, because they are not average Internet users, they were deliberately left out of this research. Another reason for excluding them was that the quality of the password, e.g., how the password is constructed, differs between students, professionals (future engineers), and average users [17].

Data were collected mainly in classrooms, as students were asked by professors, often before their lectures, to fill out online questionnaires. The link was shared through the official communication channels for teaching materials.

The students had a median age of 19, an interquartile range of 18 to 20, and a total range of 17 to 38 years old. There were 21.4% male students, 42.1% had some training regarding information security awareness, and 85.4% had self-assessed their knowledge of information security and privacy as good or excellent.

3. RESULTS AND DISCUSSION

Two main results, password revealing and not reading a statement of consent, present a relatively high level of risk in average users' online behaviour. Thus, out of 278 students, even 158 (56.8%) had written down and revealed their passwords in replying to the trick question on password quality. Because there were obvious false passwords among them (e.g., No, I will not, 123456, and similar), after the personal assessment of each answer, the number of "presumably real and discovered" passwords was reduced to 63 (22.7%). The assessment of each answer, the false password evaluation, was done as a consensus of experts, the authors of this research. However, on the new trick question regarding giving consent for data processing for research purposes, only 30 (10.8%) indicated how it was in question and requested (both to accept and decline) and, in that way, confirmed that they had read the consent. Among others, 38 (13.7%) students declined consent but continued to answer other questions and finished the whole questionnaire, while most examinees (210, 75.5%) gave their consent obviously without carefully reading it first. Here, it can be assumed that examinees may feel the false security of authority, just like when giving out the password - at the university under the supervision of the professor.

Revealing passwords and giving consent without reading the terms and conditions are two actions that can be considered hazardous online behaviour. As the P value is close to the significance level, there could be a potential correlation between these two risky actions among average online users, meaning that online users who reveal passwords usually do not read the Statement (Table 1). In total, 60 (21.6%) examinees did both risky actions. In further analyses, they were compared to the other examinees regarding all examined variables (Table 2).

Table 1. Comparison between revealing password and not reading statement

	Read statement	Didn't read statement	Total	*P**
Didn't reveal password	27 (90.0)	188 (75.8)	215 (77.3)	0.079
Revealed password	3 (10.0)	60 (24.2)	63 (22.7)	
Total	**30 (100.0)**	**248 (100.0)**	**278 (100.0)**	

*Chi-square Test

Table 2. Differences between most risky examinees and the others

Examined variable with categories		Didn't read statement and revealed password /*n*=60	Others /*n*=218	*P*
Gender/*n*(%)	male	8 (13.3)	52 (23.9)	0.079*
	female	52 (86.7)	166 (76.1)	
Age/median (25%-75%)		19 (19.0 - 21.0)	19 (18.0 - 20.0)	***0.016*** ***
Self-assessed knowledge on security and privacy/*n*(%)	poor	13 (21.7)	28 (12.8)	0.119*
	good	43 (71.7)	161 (73.9)	
	excellent	4 (6.7)	29 (13.3)	
Previous training on security/*n*(%)	Yes	26 (43.3)	90 (41.3)	0.776
	No	34 (56.7)	128 (58.7)	
Notifications from third-party partners about similar studies/*n*(%)	Yes	12 (20.0)	25 (11.5)	0.085
	No	48 (80.0)	193 (88.5)	
Receiving free anti-virus software from third-party partners/*n*(%)	Yes	22 (36.7)	60 (27.5)	0.169
	No	38 (63.3)	158 (72.5)	
Personal email address left /*n*(%)	Yes	20 (33.3)	51 (23.4)	0.118
	No	40 (66.7)	167 (76.6)	
Self-assessed risky of online behavior***/median (25%-75%)		1.0 (1.0 - 1.3)	1.3 (1.0 - 1.5)	***0.030*** ***
Cognitive importance of safe online usage/median (25%-75%)		4 (3.5 - 4.5)	4 (3.5 - 4.4)	0.595**
Cognitive awareness of online risks/median (25%-75%)		4.2 (2.8 - 4.8)	4.4 (3.2 - 4.8)	0.189**

*Chi-square Test | **Mann-Whitney Test | ***Higher score means riskier behavior

On the other three questions from the Simulation subscale, 37 (13.3%) examinees answered positively regarding receiving notifications from third-party partners about similar studies, and 82 (29.5%) answered positively regarding receiving free antivirus software from third-party partners via email. Personal email addresses were left by 71 (25.5%) of all examinees in order to receive notifications and free promotional materials.

Students who did not read the Statement and revealed the password, which is a risky action, are significantly older (Mann-Whitney test, P=0.016) than other students. However, as the absolute value of the difference is not high, maybe this result is not that important. A significant result is a significant difference in self-assessed risk of online behaviour (Mann-Whitney test, P=0.030), meaning that contradictory students that behave riskier self-assess themselves as significantly safer in online behaviour. Generally, students who behave riskier are worse in all other examined variables, except in evaluating the importance of safe online usage, even though this finding lacks statistical significance (Table 2).

The additional aim of this study was to upgrade the first subscale of the BCISQ questionnaire that measures the risk of actual online behaviour by simulating some risky online situations. The plan was to change the existing trick question on password disclosure with the new trick question on giving consent without reading the terms and conditions. Here are the results concerning Cronbach's alpha coefficient, which measures the internal consistency and reliability of a set of survey items, in this case, questions constructing a simulation subscale (Table 3).

Table 3. Differences in internal consistency regarding items of simulation subscale

Steps in statistical analysis	Number of items constructing subscale	Cronbach's alpha coefficient*	Effect of dropping variable
Step one	Four items (initial version from previous studies)	0.6812	revealing password causes change of +0.1288
Step two	Five items (added trick question on giving consent)	0.6192	giving consent, change of +0.06199 revealing password, change of +0.05677
Step three	Four items (with trick question on giving consent instead of revealing password question)**	0.6760	giving consent causes change of +0.1340
Finale step	Three items (excluded both trick questions)	0.8100	(best result)

*coefficient needs to be > 0.7 | **aim was to switch two trick questions

Even though this analysis aims to switch the existing trick question on revealing a password with the new trick question on accepting consent without reading it, the first step analysed the simulation subscale's version from the previously validated and used version of the BCISQ questionnaire. The result in step one in the table confirms that this subscale needs to be corrected and upgraded, as shown in the previous study [33] - Cronbach's alpha coefficient is lower than 0.7. The effect of dropping the trick question will increase the value of the coefficient (Table 3).

Adding a new trick question further reduces the value of Cronbach's alpha coefficient. In contrast, dropping each trick question will positively affect the internal consistency and reliability of the simulation subscale. The analysis result in step three additionally confirms that the new trick question on accepting conditions without reading them does not contribute to the internal consistency and reliability of the simulation subscale and thus needs to be dropped. However, the final step of Cronbach's alpha coefficient analysis shows an outstanding result, meaning that the internal consistency and reliability of the simulation scale are best if only three items are included. So, the result is to exclude both trick questions and construct a simulation scale with only three previously existing questions regarding receiving notifications, free antivirus, and revealing a personal email address.

4. CONCLUSIONS

The revealing of passwords and the giving of consent without reading applicable terms and conditions are two actions that could be considered extremely risky online behaviour, according to the primary results (22.7% of users revealing their passwords and even 89.2% not reading the terms and conditions), it can be concluded that behaviour is still quite risky among online users. The main result, showing a contradiction between the self-assessed and measured real risk of online behaviour, further highlights this problem. The result shows that users who behave riskier self-assess themselves as performing significantly better in risky online behaviour than they do. Users who engage in risky behaviour think they are acting safely online.

This unexpected result draws a conclusion that can be very important to information security managers and cyber security trainers. It shows that special care needs to be directed towards self-confident users, as they behave in a riskier manner when dealing with digital online data.

It seems that this particular, statistically significant result is new and not comparable but is additional information to the other empirical studies on this subject, mentioned previously in the Introduction section.

Results concerning the additional aim of this study have shown that authors were unsuccessful in replacing the old trick question asking for a password with the new trick question regarding giving consent when not reading terms and conditions. However, concerning internal consistency and reliability, the result of the simulation subscale is to reduce the subscale on three existing items presenting questions.

That is another unexpected result, but it implies a new and better simulation subscale than the previous version. So, the additional result of this empirical study is a new, improved version of the Behavioral Cognitive Internet Security Questionnaire.

Even though students were from different university faculties, excluding engineers as untypical online users, it is incorrect to conclude that these results can apply to the average online user. Another drawback of this study is the relatively small sample size, constructed only of students and only from students in their lower years of study.

Potential future research should examine all kinds of users to evaluate the average user's level of risk in online behaviour, as information security awareness is still low. Another highly beneficial research would be a review article of all the existing empirical studies on information security and privacy protection, focusing on users' awareness, knowledge and behaviour.

5. REFERENCES

[1] S. Goel, K. Williams, E. Dincelli, "Got Phished? Internet Security and Human Vulnerability", Journal of the Association for Information Systems, Vol. 18, No. 1, 2017, pp. 22-44.

[2] D. R. Vuţă, E. Nichifor, O.M. Tierean, "Extending the Frontiers of Electronic Commerce Knowledge through Cybersecurity", Electronics, Vol. 11, No. 14, 2022, p. 2223.

[3] J. Zhen, K. Dong, Z. Xie, L. Chen, "Factors Influencing Employees' Information Security Awareness in the Telework Environment", Electronics, Vol. 11, No. 21, 2022, p. 3458.

[4] H. Aldawood, G. Skinner, "Reviewing Cyber Security Social Engineering Training and Awareness Programs - Pitfalls and Ongoing Issues", Future Internet, Vol. 11, No. 3, 2019, p. 73.

[5] T. Velki, "Psychologists as information-communication system users: Is this bridge between information-communication and behavioural science enough to prevent risky online behaviours?", Proceedings of the 45th Jubilee International Convention on Information, Communication and Electronic Technology, Opatija, Croatia, 23-27 May 2022, pp. 1048-1052.

[6] N. Gerber, P. Gerber, M. Volkamer, "Explaining the privacy paradox: A systematic review of the literature investigating privacy attitude and behaviour", Computers & Security, Vol. 77, 2018, pp. 226-261.

[7] A. Bostan, I. Akman, "Impact of education on security practices in ICT", Tehnički Vjesnik - Technical Gazette, Vol. 22, No. 1, 2015, pp. 161-168.

[8] I. Borić-Letica, "Some Correlates of Risky User Behavior and ICT Security Awareness of Secondary

School Students", International Journal of Electrical and Computer Engineering Systems, Vol. 10, No. 2, 2019, pp. 85-89.

[9] A. Tick, D. J. Cranfield, I. M. Venter, "Comparing Three Countries' Higher Education Students' Cyber Related Perceptions and Behaviours during COVID-19", Electronics, Vol. 10, No. 22, 2021, p. 2865.

[10] A. R. Gillam, W. T. Foster, "Factors affecting risky cybersecurity behaviours by U.S. workers: An exploratory study", Computers in Human Behavior, Vol. 108, 2020.

[11] L. Hadlington, "Employees Attitudes towards Cyber Security and Risky Online Behaviours: An Empirical Assessment in the United Kingdom", International Journal of Cyber Criminology, Vol. 12, No. 1, 2018, pp. 269-281.

[12] I. Ortiz-Garces, R. Gutierrez, D. Guerra, "Development of a Platform for Learning Cybersecurity Using Capturing the Flag Competitions", Electronics, Vol. 12, No. 7, 2023, p. 1753.

[13] M. Amanowicz, M. Kamola, "Building Security Awareness of Interdependent Services, Business Processes, and Systems in Cyberspace", Electronics, Vol. 11, No. 22, 2022, p. 3835.

[14] V. Taneski, M. Heričko, B. Brumen, "Systematic Overview of Password Security Problems", Acta Polytechnica Hungarica, Vol. 16, No. 3, 2019, pp. 143-165.

[15] E. Y. Güven, A. Boyaci, M. A. Aydin, "A Novel Password Policy Focusing on Altering User Password Selection Habits: A Statistical Analysis on Breached Data", Computers & Security, Vol. 113, 2022, p. 102560.

[16] M. Curry, B. Marshall, J. Correia, R. E. Crossler, "InfoSec Process Action Model (IPAM): Targeting Insiders' Weak Password Behavior", Journal of Information Systems, Vol. 33, No. 3, 2019, pp. 201-225.

[17] R. Alomari, J. Thorpe, "On password behaviours and attitudes in different populations", Journal of Information Security and Applications, Vol. 45, 2019, pp. 79-89.

[18] W. Albattah, "Analysis of passwords: Towards an understanding of strengths and weaknesses", International Journal of Advanced And Applied Sciences, Vol. 5, No. 11, 2018, pp. 51-60.

[19] A. Juozapavičius, A. Brilingaitė, L. Bukauskas, R. G. Lugo, "Age and Gender Impact on Password Hygiene", Applied Sciences, Vol. 12, No. 2, 2022, p. 894.

[20] J. P. Kaleta, J. S. Lee, S. Yoo, "Nudging with construal level theory to improve online password use and intended password choice", Information Technology & People, Vol. 32, No. 4, 2019, pp. 993-1020.

[21] T. Velki, K. Romstein, "User Risky Behavior and Security Awareness through Lifespan", International Journal of Electrical and Computer Engineering Systems, Vol. 9, No. 2, 2018, pp. 53-60.

[22] M. A. El-Sayed, M. A. Abdel-Latif, "Achieving Information Security by multi-Modal Iris-Retina Biometric Approach Using Improved Mask R-CNN", International Journal of Electrical and Computer Engineering Systems, Vol. 14, No. 6, 2023, pp. 657-665.

[23] N. Balan, V. Ila, "A Novel Biometric Key Security System with Clustering and Convolutional Neural Network for WSN", Tehnicki Vjesnik - Technical Gazette, Vol. 29, No. 5, 2022, pp. 1483-1490.

[24] V. Thakkar, V. Shah, "A Privacy-Preserving Framework Using Hyperledger Fabric for EHR Sharing Applications", International Journal of Electrical and Computer Engineering Systems, Vol. 14, No. 6, 2023, pp. 667-676.

[25] T. Perera, T. Perera, "Barrister-Processing and Summarisation of Terms & Conditions / Privacy Policies", Proceedings of the 6th International Conference for Convergence in Technology, Maharashtra, India, 2-4 April 2021, pp. 1-7.

[26] P. Martiskova, R. Svec, M. Slaba, "Online Shopping and Reading E-Shops' Terms and Conditions", Education Excellence and Innovation Management through Vision 2020, Proceedings of the 33rd International Business Information Management Association Conference, Granada, Spain, 10-11 April 2019, pp. 682-690.

[27] J. A. Obar, A. Oeldorf-Hirsch, "The biggest lie on the Internet: ignoring the privacy policies and

terms of service policies of social networking services", Information, Communication & Society, Vol. 23, No. 1, 2020, pp. 128-147.

[28] N. Steinfeld, "I agree to the terms and conditions": (How) do users read privacy policies online? An eye-tracking experiment", Computers in Human Behavior, Vol. 55, Part B, 2016, pp. 992-1000.

[29] T. Velki, K. Šolić, "Development and Validation of a New Measurement Instrument: The Behavioral-Cognitive Internet Security Questionnaire (BCISQ)" International Journal of Electrical and Computer Engineering Systems, Vol. 10, No. 1, 2019, pp. 19-24.

[30] S. Egelman, M. Harbach, E. Peer, "Behavior ever follows intention? A validation of the security behaviour intentions scale (SeBIS)", Proceedings of the CHI Conference on Human Factors in Computing Systems, New York, NY, USA, May 2016.

[31] G. Öğütçü, Ö. M. Testik, O. Chouseinoglou, "Analysis of personal information security behaviour and awareness", Computer Security, Vol. 56, 2016, pp. 83-93.

[32] K. Parsons, D. Calic, M. Pattinson, "The Human Aspects of Information Security Questionnaire (HAIS-Q): Two further validation studies", Computer Security, Vol. 66, 2017, pp. 40-51.

[33] T. Velki, K. Šolić, B. Žvanut, "Cross-cultural validation and psychometric testing of the Slovenian version of the Croatian Behavioral-Cognitive Internet Security Questionnaire", Elektrotehniški Vestnik, Vol. 89, No. 3, 2022, pp. 103-108

INTERNATIONAL JOURNAL OF ELECTRICAL AND COMPUTER ENGINEERING SYSTEMS

Published by Faculty of Electrical Engineering, Computer Science and Information Technology Osijek,
Josip Juraj Strossmayer University of Osijek, Croatia.

About this Journal

The International Journal of Electrical and Computer Engineering Systems publishes original research in the form of full papers, case studies, reviews and surveys. It covers theory and application of electrical and computer engineering, synergy of computer systems and computational methods with electrical and electronic systems, as well as interdisciplinary research.

Topics of interest include, but are not limited to:

- Power systems
- Renewable electricity production
- Power electronics
- Electrical drives
- Industrial electronics
- Communication systems
- Advanced modulation techniques
- RFID devices and systems
- Signal and data processing
- Image processing
- Multimedia systems
- Microelectronics
- Instrumentation and measurement
- Control systems
- Robotics
- Modeling and simulation
- Modern computer architectures
- Computer networks
- Embedded systems
- High-performance computing
- Parallel and distributed computer systems
- Human-computer systems
- Intelligent systems
- Multi-agent and holonic systems
- Real-time systems
- Software engineering
- Internet and web applications and systems
- Applications of computer systems in engineering and related disciplines
- Mathematical models of engineering systems
- Engineering management
- Engineering education

Paper Submission

Authors are invited to submit original, unpublished research papers that are not being considered by another journal or any other publisher. Manuscripts must be submitted in doc, docx, rtf or pdf format, and limited to 30 one-column double-spaced pages. All figures and tables must be cited and placed in the body of the paper. Provide contact information of all authors and designate the corresponding author who should submit the manuscript to https://ijeces.ferit.hr. The corresponding author is responsible for ensuring that the article's publication has been approved by all coauthors and by the institutions of the authors if required. All enquiries concerning the publication of accepted papers should be sent to ijeces@ferit.hr.

The following information should be included in the submission:

- paper title;
- full name of each author;
- full institutional mailing addresses;
- e-mail addresses of each author;
- abstract (should be self-contained and not exceed 150 words). Introduction should have no subheadings;
- manuscript should contain one to five alphabetically ordered keywords;
- all abbreviations used in the manuscript should be explained by first appearance;
- all acknowledgments should be included at the end of the paper:
- authors are responsible for ensuring that the information in each reference is complete and accurate. All references must be numbered consecutively and citations of references in text should be identified using numbers in square brackets. All references should be cited within the text;
- each figure should be integrated in the text and cited in a consecutive order. Upon acceptance of the paper, each figure should be of high quality in one of the following formats: EPS, WMF, BMP and TIFF;
- corrected proofs must be returned to the publisher within 7 days of receipt.

Peer Review

All manuscripts are subject to peer review and must meet academic standards. Submissions will be first considered by an editor-in-chief and if not rejected right away, then they will be reviewed by anonymous reviewers. The submitting author will be asked to provide the names of 5 proposed reviewers including their e-mail addresses. The proposed reviewers should be in the research field of the manuscript. They should not be affiliated to the same institution of the manuscript author(s) and should not have had any collaboration with any of the authors during the last 3 years.

Author Benefits

The corresponding author will be provided with a .pdf file of the article or alternatively one hardcopy of the journal free of charge.

Units of Measurement

Units of measurement should be presented simply and concisely using System International (SI) units.

Bibliographic Information

Commenced in 2010.
ISSN: 1847-6996
e-ISSN: 1847-7003

Published: semiannually

Copyright

Authors of the International Journal of Electrical and Computer Engineering Systems must transfer copyright to the publisher in written form.

Subscription Information

The annual subscription rate is 50€ for individuals, 25€ for students and 150€ for libraries.

Postal Address

Faculty of Electrical Engineering,
Computer Science and Information Technology Osijek,
Josip Juraj Strossmayer University of Osijek, Croatia
Kneza Trpimira 2b
31000 Osijek, Croatia

IJECES Copyright Transfer Form

(Please, read this carefully)

This form is intended for all accepted material submitted to the IJECES journal and must accompany any such material before publication.

TITLE OF ARTICLE (hereinafter referred to as "the Work"):

COMPLETE LIST OF AUTHORS:

The undersigned hereby assigns to the IJECES all rights under copyright that may exist in and to the above Work, and any revised or expanded works submitted to the IJECES by the undersigned based on the Work. The undersigned hereby warrants that the Work is original and that he/she is the author of the complete Work and all incorporated parts of the Work. Otherwise he/she warrants that necessary permissions have been obtained for those parts of works originating from other authors or publishers.

Authors retain all proprietary rights in any process or procedure described in the Work. Authors may reproduce or authorize others to reproduce the Work or derivative works for the author's personal use or for company use, provided that the source and the IJECES copyright notice are indicated, the copies are not used in any way that implies IJECES endorsement of a product or service of any author, and the copies themselves are not offered for sale. In the case of a Work performed under a special government contract or grant, the IJECES recognizes that the government has royalty-free permission to reproduce all or portions of the Work, and to authorize others to do so, for official government purposes only, if the contract/grant so requires. For all uses not covered previously, authors must ask for permission from the IJECES to reproduce or authorize the reproduction of the Work or material extracted from the Work. Although authors are permitted to re-use all or portions of the Work in other works, this excludes granting third-party requests for reprinting, republishing, or other types of re-use. The IJECES must handle all such third-party requests. The IJECES distributes its publication by various means and media. It also abstracts and may translate its publications, and articles contained therein, for inclusion in various collections, databases and other publications. The IJECES publisher requires that the consent of the first-named author be sought as a condition to granting reprint or republication rights to others or for permitting use of a Work for promotion or marketing purposes. If you are employed and prepared the Work on a subject within the scope of your employment, the copyright in the Work belongs to your employer as a work-for-hire. In that case, the IJECES publisher assumes that when you sign this Form, you are authorized to do so by your employer and that your employer has consented to the transfer of copyright, to the representation and warranty of publication rights, and to all other terms and conditions of this Form. If such authorization and consent has not been given to you, an authorized representative of your employer should sign this Form as the Author.

Authors of IJECES journal articles and other material must ensure that their Work meets originality, authorship, author responsibilities and author misconduct requirements. It is the responsibility of the authors, not the IJECES publisher, to determine whether disclosure of their material requires the prior consent of other parties and, if so, to obtain it.

- The undersigned represents that he/she has the authority to make and execute this assignment.
- For jointly authored Works, all joint authors should sign, or one of the authors should sign as authorized agent for the others.
- The undersigned agrees to indemnify and hold harmless the IJECES publisher from any damage or expense that may arise in the event of a breach of any of the warranties set forth above.

______________________________ ______________________________

Author/Authorized Agent **Date**

CONTACT
International Journal of Electrical and Computer Engineering Systems (IJECES)
Faculty of Electrical Engineering, Computer Science and Information Technology Osijek
Josip Juraj Strossmayer University of Osijek
Kneza Trpimira 2b
31000 Osijek, Croatia
Phone: +38531224600,
Fax: +38531224605,
e-mail: ijeces@ferit.hr