Princess Sumaya University for Technology
جامعة الأميرة سمية للتكنولوجيا

صندوق دعم البحث العلمي والابتكار
Scientific Research and Innovation Support Fund

Jordanian Journal of Computers and Information Technology

September 2022 VOLUME 08 NUMBER 03
ISSN 2415 - 1076 (Online)
ISSN 2413 - 9351 (Print)

JJCIT

www.jjcit.org jjcit@psut.edu.jo

An International Peer-Reviewed Scientific Journal Financed by the Scientific Research and Innovation Support Fund

Jordanian Journal of Computers and Information Technology (JJCIT)

The Jordanian Journal of Computers and Information Technology (JJCIT) is an international journal that publishes original, high-quality and cutting edge research papers on all aspects and technologies in ICT fields.

JJCIT is hosted and published by Princess Sumaya University for Technology (PSUT) and supported by the Scientific Research Support Fund in Jordan. Researchers have the right to read, print, distribute, search, download, copy or link to the full text of articles. JJCIT permits reproduction as long as the source is acknowledged.

AIMS AND SCOPE

The JJCIT aims to publish the most current developments in the form of original articles as well as review articles in all areas of Telecommunications, Computer Engineering and Information Technology and make them available to researchers worldwide. The JJCIT focuses on topics including, but not limited to: Computer Engineering & Communication Networks, Computer Science & Information Systems and Information Technology and Applications.

INDEXING

JJCIT is indexed in:

ACADEMIA


EDITORIAL BOARD SUPPORT TEAM

LANGUAGE EDITOR

Haydar Al-Momani

EDITORIAL BOARD SECRETARY

Eyad Al-Kouz


All articles in this issue are open access articles distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/4.0/).

JJCIT ADDRESS

WEBSITE: www.jjcit.org
EMAIL: jjcit@psut.edu.jo
ADDRESS: Princess Sumaya University for Technology, Khalil Saket Street, Al-Jubaiha
B.O. BOX: 1438 Amman 11941 Jordan
TELEPHONE: +962-6-5359949
FAX: +962-6-7295534

EDITORIAL BOARD

Wejdan Abu Elhaija (EIC)
Ahmad Hiasat (Senior Editor)
Aboul Ella Hassanien
Adil Alpkoçak
Adnan Gutub
Adnan Shaout
Christian Boitet
Gian Carlo Cardarilli
Omer Rana
Abdelfatah Tamimi
Nijad Al-Najdawi
Hussein Al-Majali
Maen Hammad
Ayman Abu Baker
Essam Al-Dawood
João L. M. P. Monteiro
Leonel Sousa
Omar Al-Jarrah

INTERNATIONAL ADVISORY BOARD

Ahmed Yassin Al-Dubai
UK
Albert Y. Zomaya
AUSTRALIA
Chip Hong Chang
SINGAPORE
Izzat Darwazeh
UK
Dia Abu Al Nadi
JORDAN
George Ghinea
UK
Hoda Abdel-Aty Zohdy
USA
Saleh Oqeili
JORDAN
João Barroso
PORTUGAL
Karem Sakallah
USA
Khaled Assaleh
UAE
Laurent-Stephane Didier
FRANCE
Lewis Mackenzies
UK
Zoubir Hamici
JORDAN
Korhan Cengiz
TURKEY
Marco Winzker
GERMANY
Marwan M. Krunz
USA
Mohammad Belal Al Zoubi
JORDAN
Michael Ullman
USA
Ali Shatnawi
JORDAN
Mohammed Benaissa
UK
Basel Mahafzah
JORDAN
Nadim Obaid
JORDAN
Nazim Madhavji
CANADA
Ahmad Al Shamali
JORDAN
Othman Khalifa
MALAYSIA
Shahrul Azman Mohd Noah
MALAYSIA
Shambhu J. Upadhyaya
USA

"Opinions or views expressed in papers published in this journal are those of the author(s) and do not necessarily reflect those of the Editorial Board, the host university or the policy of the Scientific Research and Innovation Support Fund".

"ما ورد في هذه المجلة يعبر عن آراء الباحثين ولا يعكس بالضرورة آراء هيئة التحرير أو الجامعة أو سياسة صندوق دعم البحث العلمي والابتكار".

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 08, No. 03, September 2022.

PHYLOGENETIC REPLAY LEARNING IN DEEP NEURAL NETWORKS

Jean-Patrice Glafkides[1], Gene I. Sher[2] and Herman Akdag[1]

(Received: 31-Jan.-2022, Revised: 18-Apr.-2022, Accepted: 22-Apr.-2022)

ABSTRACT

Though substantial advancements have been made in training deep neural networks, one problem remains, the vanishing gradient. The very strength of deep neural networks, their depth, is also unfortunately their problem, due to the difficulty of thoroughly training the deeper layers due to the vanishing gradient. This paper proposes "Phylogenetic Replay Learning", a learning methodology that substantially alleviates the vanishing-gradient problem. Unlike the residual learning methods, it does not restrict the structure of the model. Instead, it leverages elements from neuroevolution, transfer learning and layer-by-layer training. We demonstrate that this new approach is able to produce a better performing model and by calculating Shannon entropy of weights, we show that the deeper layers are trained much more thoroughly and contain statistically significantly more information than when a model is trained in a traditional brute force manner.

KEYWORDS

Neural networks, Neuroevolution, Phylogenetic replay learning, Deep learning, Vanishing gradient.

1. INTRODUCTION

Nature evolved the nervous system through eons of trial and error, from the first apparition of the neuronal cell to the complex brains we possess today. The field of machine learning has made tremendous progress during the past decade, predominantly owing to the improvement of CPU performance, data accessibility, optimization of deep neural network (DNN) algorithms, but also just as significantly due to the improvements in hardware and the use of GPUs. Artificial neural networks are called deep when they have more than 3 layers of neurons (though some categorize DNNs as those having more than 9 layers) and are capable of being tuned to reach a specific goal through the use of an optimization algorithm, mimicking the role of synaptic plasticity in biological learning. This approach has led to the emergence of highly efficient algorithms that are capable of learning and solving complex problems [1]. Two of the main limitations of such algorithms are: 1. Their topologies are built empirically and 2. Due to the depth of deep neural networks, they are affected by the vanishing-gradient problem. Though this paper primarily concentrates on solving the 2nd problem (the vanishing-gradient problem), we demonstrate its use by applying it to a model that was evolved through neuroevolution.

We do this because:

1. In the last few years, substantial advancements have been made in automated model search and construction. These automated model construction and model search methods are commonly called neuroevolutionary methods, due to the use of evolutionary algorithms to search for optimal model architectures [2]. These methods have demonstrated a strong ability to produce state-of-the-art models demonstrating excellent results in numerous domains [3]-[5] with very surprising results in some cases [6]-[7]. Several works exploring the use of evolutionary computation in deep network optimization [8]-[10] were produced.

2. Our new proposed method, Phylogenetic Replay Learning (PRL), can be perfectly combined with both, traditional, but also neuroevolutionary methods to leverage the ability to construct deep and complex networks from simple ones.

It must be noted that the objective of this paper is not to discuss or compare any specific model search or neuroevolutionary method, like EANT1/2 [11], CoSYnE [12], DXNN or NEAT, efficiency over other methods that have already been addressed [13]-[15], but to explore the use of backpropagation training in pre-planned mutations, training layers one at a time as the deep neural network is constructed. With all the accomplishments of deep learning, it remains difficult to build models that generalize or adapt

1. J.-P. Glafkidès (ORCHID: 0000-0001-5273-9948) and H. Akdag are with PARAGRAPHE EA 349 - PARIS VIII University, France. Emails: jp@glafonline.com and herman.akdag@univ-paris8.fr

2. G. I. Sher (ORCHID: 0000-0002-2086-4370) is with DataValoris - WY, USA. Email: gene@datavaloris.com

efficiently to complex-problem domains and data. One of the bigger difficulties being faced when building complex and deep models that converge correctly is the vanishing-gradient problem [16]-[18] which is yet to be solved [19]. It is this problem, the vanishing gradient, that the PRL approach is also aimed at solving. With the increasing number of layers that are used, the vanishing-gradient problem can cause the gradient to become too small for effective weight parameter updating. This is due to certain activation functions, like the sigmoid function, which squashes a large input space into a small one between 0 and 1. Thus, a large change in the input of the sigmoid function will cause a small change in the output and with it the derivative also shrinks. This problem is exacerbated with deeper layering; the gradient decreases exponentially as we propagate down to the initial layers. A small gradient means that the weights and biases of the initial (deeper) layers will not be trained effectively. Since these initial layers are often crucial to recognizing the core elements of the input data, this can lead to overall inability of the whole network to learn effectively. This effect can be partially mitigated by using other activation functions, such as relu for example. Other ways of combating this problem are specific architectures, like the residual neural network [20] which attempts to decrease the effect of this problem by connecting deeper layers directly to the output. However, it is not enough and too restrictive. This calls for the development of new methods specifically designed to enhance learning capabilities and counter the vanishing-gradient effect. A method is needed that will not restrict us to the use of specific neural topologies or activation functions.

The objective of this paper is to compare the performance of training a DNN all at once, *versus* training it one mutation at a time as a pre-planned model is being constructed (PRL training) and demonstrate that the latter produces a better outcome, with each layer of such model storing statistically greater amount of information. The Phylogenetic Replay Learning (PRL) requires a trace of model's complexification, from a simple shallow version to the final complex DNN. When this trace is available, it performs re-training of the layers as it adds layer on-top of layer within the trace. This iterative re-training approach ensures that every layer was at some point the output layer (or close to it) and thus was affected by the gradient descent learning algorithm to a greater extent, while the deeper layers were "re-tuned" to work effectively in the deeper model. When this approach is combined with neuroevolution, the system first evolves the final model from a simple initial seed model while also building its trace of mutations (which new layers are added on top of which or which layer is changed or get linked to others) and then it re-traces those evolutionary steps (the phylogeny), while re-training the model at every evolutionary step, as shown in the Figure 1. In the following sections, we will discuss in detail the PRL method. First, we will cover the background of the pertinent domains, neuroevolution and the vanishing-gradient problem. We will then provide definitions of the terms used in this paper. In the methods section, we provide a detailed PRL algorithm. In the results section, we will present the experiments performed and their results. Finally, we will conclude with the analysis and discussion of the results achieved.

2. BACKGROUND

2.1 The Vanishing-gradient Effect (VGE)

The most common neural network (NN) optimization algorithm is based on the use of stochastic gradient descent. This involves first calculating the prediction error made by the model and then using the error to estimate a gradient used to update each weight layer by layer, cascading backwards in the network. This error gradient is propagated backward through the network from the output layer to the input layer, updating the weights to minimize the difference between the actual NN output and the expected output. It is useful to train NNs with many layers. The addition of deeper layers increases its capacity, making it capable of learning more complex mapping functions between input and output when a large training dataset is provided. A problem with training networks with many layers (e.g. deep neural networks) is that the gradient diminishes dramatically as it is propagated backward through the network. The error may be so small by the time it reaches layers close to the input of the model that it may have very little effect. Thus, this problem is referred to as the “vanishing-gradient” problem.

2.2 Neuroevolution

Neuroevolution is a machine-learning technique that applies an evolutionary algorithm to construct artificial NNs, taking inspiration from the biological evolutionary process.

2.3 Definitions

Champion: is an NN model (topology and weights) representing the best model that neuroevolution is able to produce to solve a problem.

Direct Deep Learning (DDL): is what we call the standard/default training of a model using backpropagation (Adam, QProp, …etc.) to differentiate it from the PRL method. It is a method that is applied to the DNN without the use of neuroevolution or PRL. In our experiments, the training algorithm used in the framework was set to Adam. The DDL is also known as end-end training

Hall of Fame (HOF): or HOF for short, is a list our neuroevolutionary system holds of the best performing agents/models. In our tests, HOF was set to size 10,

Initial Model: is the seed model used as the starting point of model search in neuroevolution.

Mutations: at each step of the evolutionary process, we apply mutation(s) to the topology of the parent in order to create an offspring. A topological mutation can add a layer to the model, mutate existing layer's parameters, remove a layer, clone an existing layer, add or change a link between two layers or swap one layer for another type of layer.

Phylogenetic Replay Learning (PRL): is a method of training a model for a specific problem using pre-recorded mutation path of a seed model topology (system implemented and presented in this paper), but doing so one mutation step at a time, following that model's phylogenetic path. In other words, we re-train the model after every applied mutation step once we know what the best model is and what mutation steps were taken to achieve it from the seed model, usually following the path of model complexification from the initial neuroevolution phases. This method is the topic of this paper.

Selection Process: is the mechanism by which the algorithm selects the best entities according to their score (fitness function) and stores them in the "Hall of Fame" (HOF) list.

2.4 Other Methods to Reduce Vanishing-gradient Effect (VGE)

Several other approaches can be used to reduce the VGE, but none are perfect. Using PRL does not preclude one from leveraging other methods as well.

- Activation functions, such as relu for example [21].
- Normalized initialization layers [22]-[23] and intermediate normalization layers [24], which enable networks with tens of layers to start learning/converging with stochastic gradient descent (SGD) with backpropagation [25].
- Specific architectures like the residual neural networks which attempt to decrease the effect of this problem by using pass-through links [20].
- Regularizing deep neural networks by noise injects noise during the training procedure, adding or multiplying noise within the hidden units of the NNs [26].
- Deep cascade learning method proposes a solution to alleviate the VGE [27] by training deep networks in a cascade-like or bottom-up layer-by-layer manner. It reduces the VGE, but was not shown to be better than DDL.

2.5 Metrics

The metrics we use for model comparison is the test accuracy. Early stopping was applied on the score we want to follow and not used for training. Accuracy is used as the metric. To better understand the difference in the informational density of the models, we calculated their weights' Shannon entropy [28] Equation 1, Equation 2 after training.

$$P_i = \frac{i}{\sum_{i=1}^{n}(i)} \tag{1}$$

H entropy:

$$H = -\sum_{i=1}^{n} P_i \ln(P_i) \tag{2}$$

2.6 Dataset

PRL was tested on the 4 "original" datasets from Keras site: MNIST, Fashion MNIST, CIFAR 10 and Tiny Imagenet. CIFAR10 was converted into grayscale with images reshaped to 28*28 pixels, to not only match the same shape as those within MNIST, but also to make it much more complex to learn.

Tiny Imagenet 200 dataset has been chosen to test the system on a more modern, bigger and more difficult to learn dataset. This paper's aim is to compare the PRL method to the standard approach. Thus, the goal of this work is to show that on average, this training approach produces better performing models, with more densely packed information, than the direct approach, by alleviating the VGE. Thus, we believe that for these preliminary results, it is appropriate to use these datasets.

2.7 Tools

We selected tools like Keras that provides the training framework and Raise solution from DataValoris that provides the evolutionary part of the experiments on top of Keras. They have accelerated our work as their engine already provides the unrestricted topological search-based deep-learning neuroevolution. The PRL (recording and replay training) was developed by us for the purpose of this work and presentation of experiments and their results in this paper. The method could be as easily used with other neuroevolutionary systems, like NEAT, EANT1/2, DXNN or GNARL, as long as we record the phylogenetic path of mutations that can then be used to replay the mutations and train the model one step at a time. Finally, all of our experiments were performed on a server with an Nvidia Tesla v100 GPU card. Part of this work was granted access to the HPC/AI resources of IDRIS under the allocation 2021- AD011012674 made by GENCI.

2.8 Seed Model

Table 1 shows the simple model used as the seed model. It includes 7850 parameters and 1 hidden layer in a sequential architecture.

Table 1. Initial model test 1.

Layer Number	Type	Output	Params
1	InputLayer	N, 28, 28, 1	0
2	Flatten	N, 784	0
3	Dense	N, 10	7850

2.9 Selection Rules

During the building of the phylogenetic path, the neuroevolutionary process uses selection based on the score generated by the learning algorithm. The score used as a fitness is the test accuracy of the model. We have set the system such that the learning rate is decreased when the score does not improve for 3 consecutive evaluations. Every generation 10 NNs are trained, then their scores are compared to the NNs in HOF. If a score of an offspring/mutant model within the current generation is higher than that of a model within the HOF that has the same topology, the mutant model replaces the model within the HOF. If the mutant model has the highest score and has a topology not present within the HOF, the model with the lowest fitness within the HOF is removed and the new model is added in its spot.

3. METHODS

PRL is a method to train models using genetically planned mutations over time. It alleviates the vanishing gradient effect through its complete training. The system allows the classical gradient descent method to train each layer, even the very deep ones, more than the traditional learning approach. It does this by retraining each of those layers as the model is being evolved and new layers are added. Each new layer added has the chance of being trained as if it were the first or second layer in the backprop cascade. Frameworks used in this study were the official TensorFlow and plaidML. Datasets have not been augmented during tests. The algorithms were developed in Python. To use the PRL algorithm the experiments have been cut into two phases

3.1 Phase 1: Generating the Champion's Mutation Path through Neuroevolution

PRL requires the existence of the phylogenetic path of the model we need to train. The first phase is meant to build the Champion model while recording its phylogenetic path (mutations that were applied sequentially to generate it). Neuroevolution is used to accomplish (Figure 1) this.

Neuroevolution generates a phylogenetic path (Figure 2) of the best performing model topology aka "champion". In this figure, the champion has 3 ancestors. The figure also shows which

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 08, No. 03, September 2022.

topological mutations were applied to get from one model to the next. The neural architecture used is built by the evolutionary process (could be CNN, Dense layers, Resnet like structure…etc.).

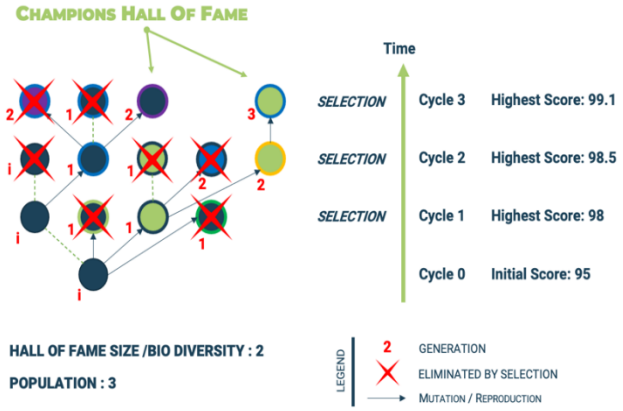


Figure 1. Selection mechanism sample.

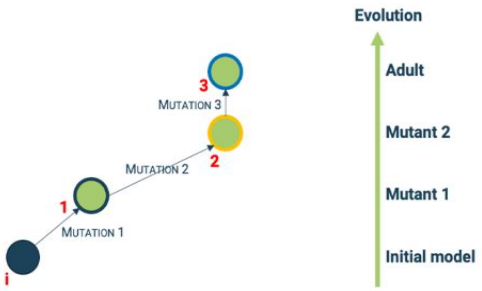


Figure 2. Phylogenetic path of champion.

3.2 Phase 2: Model Generation with the PRL

Now that we have a phylogenetic path that leads to the champion model, we can replay the path from the seed model to champion model (Figure 3).

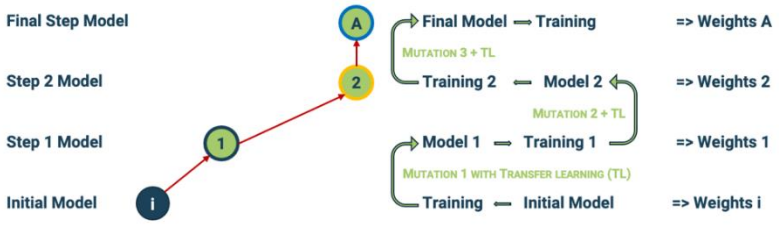


Figure 3. The phylogenetic learning path from initial model to the final one.

When replaying the phylogenetic path, we have to: 1. Generate the seed model with a new set of random synaptic weights and 2. Generate random weights when adding new layers during mutations. This then creates the final model with the same topology as the champion model, but with its own set of parameters. This is a way to statistically include in testing the impact of the initial random weights.

This study was composed of the following steps:

- Phylogenetic path recording: First, an initial simple seed model is trained on a dataset. Using the neuroevolutionary approach, over multiple generations a more complex and better performing NN architecture is evolved and the evolutionary steps leading from the seed NN to the final architecture are recorded in its mutation trace list. The final architecture is what we call the champion model.
- PRL training evaluation: Having the trace from the initial seed model to the champion, the seed model is re-trained using the PRL method X# of times. Using the PRL method, after the application of each mutation in the mutation trace, the system is retrained. This is done for every mutation step, from seed to champion, without resetting the weights between each mutation/training step (in some sense, similarly to transfer learning). This provided the average performance (average of X # of times) of the same champion topology, but trained using the PRL method.
- Champion model DDL retraining: The champion model was re-initialized with random weights and trained on the dataset X# of times using the standard learning approach. This was done to calculate the average performance of the model trained in the standard manner (to which we refer in this paper as "directly applied deep learning" or DDL), with different initial synaptic weights. The early stop patience and epoch number were set to 9 and 60 to avoid a bias where the DDL might not have enough time to train very deep networks.
- Reproducibility testing: In order to confirm the results and test the reproducibility of the method, we did the experiment more than once and on different frameworks. Another champion was

created and PRL again applied using a new seed model, another framework as well as applying it to the more complex Tiny ImageNet dataset.

- Transferability testing: We were also interested in whether the generated model was generalizable to other problems from the same domain and the difference between DDL and PRL-based methods when it comes to transferability. To evaluate the transferability of the Model using the PRL process, we also tested the same champion on other datasets, by retraining it using DDL and PRL methods.
- Data-storage efficiency testing: We calculated the efficiency of information storage in complex models trained through PRL and compared the results to those trained with DDL.

4. RESULTS OF EXPERIMENT 1

In this section, we will first generate a champion and then store its phylogenetic path. Then, we will replay the recorded mutation path with the resulting statistics and compare them to the DDL results.

4.1 Phase 1: Champion 1 Generation

The experiment was setup as follows:

- When using the neuroevolutionary method, a seed population of 20 random minimalistic models is generated.
- 20 agents are generated during every cycle (by way of mutation) from the best agents within the HOF (with a HOF max size of 10), where the probability of using any one agent as the parent of the mutant offspring being proportional to its relative fitness (accuracy) as compared to other HOF agents.
- This experiment used the MNIST dataset.
- The evolutionary engine applied 1-2 (randomly chosen) mutations to create a mutant offspring model from the parent.

The deep-learning parameters used were as follows: 20 epochs with early stopping based on a patience of 3, where patience is based on the test loss metric.

Point of attention: In this work, we refer to the "number of parents since origin" as the agents' generation number. In classic genetic algorithms, the generation is what in this study we call "cycles", therefore an agent of generation 3 and cycle 8 means that it appeared on the 8th iteration and has 3 ancestors (it could have appeared at minimum between cycles 3 to 8).

From the list of champions generated using the neuroevolutionary method during phase 1, we chose the best one, as shown in Table 2.

Table 2. Champion 1 results' information (MNIST).

Score	Cycle	Generation	Parameters	Nodes	Layers
0.9944	96	19	409158	25	13

The chosen champion has 409158 parameters spread between 25 nodes that are 13 layers deep. It has been generated on the 96th cycle and is generation 19 (it has 19 ancestors). Its score 99.44% is close to state-of-the-art on non-augmented MNIST dataset. The mutations recorded at each step that lead to the final champion topology are displayed in Table 3. At every evolutionary step, 1-2 mutation(s) were applied. The number of mutations applied at each step is limited to a maximum of 2 in order to generate a complex model with small changes between each step, which allows PRL to work on smaller parts during each mutation.

Table 3 presents a base of comparison; it shows PRL scores of the champion NN at each step of its evolutionary path. Those scores have been used as the selection criteria for HOF entrance of the offsprings during the evolutionary process. This first result shows that the model has increased in size. This is a classic behavior of an evolutionary algorithm if no size restrictions are used during model generation and mutation. We also see that the Shannon entropy decreases from generation to generation, from 8.98 to 8.90 (excluded initial model of 12.51).

We can interpret this reduction as the increase in organization and amount of useful information stored

by the model's weights. We move from an almost random set of weights to a set of weights that store useful information, a more organized distribution.

Table 3. Phylogenetic path and scores of the chosen champion.

STEP	SIZE	SCORE	SHANNON	STEP	SIZE	SHANNON	SCORE
0	7850	8.79	12.5151	10	264970	8.92824	99.3
1	94906	97.51	8.98112	11	269130	8.92447	99.27
2	27082	98.43	8.97188	12	300874	8.92426	99.28
3	58538	98.96	8.98559	13	300874	8.91894	99.33
4	90346	99.03	8.98102	14	304970	8.91918	99.34
5	90282	99.03	8.96616	15	304970	8.91798	99.34
6	183818	99.23	8.95914	16	304970	8.91156	99.35
7	183818	99.19	8.9567	17	304970	8.90396	99.36
8	258570	99.24	8.94914	18	405486	8.90111	99.41
9	264330	99.29	8.9393	19	409158	8.90037	99.44

4.2 Phase 2: DDL *versus* PRL Statistics

During phase 2, we gather the result metrics of the two different learning approaches to evaluate the impact of using PRL as compared to DDL.

4.2.1 DDL of Champion 1

To evaluate the learning capacity of the model, we conducted 50 runs using the standard learning method applied directly to the final champion model. The initial weights in each experiment were randomly generated. This number of runs allows us to calculate a statistically relevant standard deviation. In theory, the DDL of the champion model could have the same performance as the original champion (and potentially higher), but the probability that these 409158 random parameters reach an optimum is very low. The more complex and deeper the model, the greater the effect PRL method is expected to produce by countering the vanishing-gradient effect (VGE). To perform these experiments and to maximize the probability of reaching a good local minimum, 60 epochs per run were used, with patience set to 6. During our experiments, a maximum of 53 epochs were used before early stoppage occurred. An average of 45 epochs out of 60 were used before early stoppage was triggered. During phase 1 of the PRL method, the champion achieved an accuracy of 99.44%. Its Shannon entropy is 8.90037. The best score/accuracy achieved using DDL of the champion model was 99.05%, with a statistically significant difference (Table 4). We suspect that the VGE is the root cause of this result. Furthermore, we can also see that Shannon entropy of the best performing model trained using the standard approach (9.1227) is also higher than the entropy of the champion model produced during phase 1 of the PRL method.

Table 4. Applying DDL to the champion model (MNIST).

	SCORE	SHANNON
BEST	99.05	9.1227
MEAN	98.93	9.1615
Standard Deviation	0.067	0.0168

We see that the application of DDL to the model is also less efficient than that produced through phase 1 of the PRL method.

4.2.2 Phylogenetic Replay Learning

From initial model, the mutations are applied based on the phylogenetic path of the champion model. The weights are randomly generated for the new mutated layers as well as seed model. We reran the PRL experiment 50 times to gather data on which to base our averages. Weights were not reset between mutations (which can be considered as transfer learning). Table 5 shows the results of the 50 PRL experiments.

The best score reached was 99.40% with an average of 99.26%. This score is very close to that of the original champion model, which reached 99.44%. Thus, there is substantial consistency. Table 6 shows statistical information of the DDL and PRL experiments.

Table 5. PRL of the champion model (MNIST).

Step	Mean Score	Std. Deviation	Best Score	Shannon	Step	Mean Score	Std. Deviation	Best Score	Shannon
0	92.14	0.0771	92.33	12.5163	10	99.16	0.0647	99.32	8.8497
1	97.68	0.2711	98.11	8.9256	11	99.17	0.0700	99.35	8.8454
2	98.35	0.0925	98.58	8.9015	12	99.18	0.0563	99.31	8.8437
3	98.88	0.0841	99.02	8.9079	13	99.19	0.0641	99.34	8.8415
4	99.01	0.0676	99.16	8.8960	14	99.19	0.0626	99.34	8.8396
5	99.05	0.0634	99.13	8.8802	15	99.19	0.0618	99.31	8.8373
6	99.12	0.0553	99.23	8.8749	16	99.24	0.0606	99.36	8.8262
7	99.11	0.0541	99.22	8.8702	17	99.24	0.0521	99.37	8.8208
8	99.14	0.0515	99.27	8.8628	18	99.24	0.0542	99.35	8.8185
9	99.14	0.0660	99.26	8.8547	19	99.26	0.0628	99.40	8.8147

Table 6. Statistics of experiments.

	DDL	PRL		
Mean Score	98.93%	99.26%	POOLED VARIANCE	4.2E-07
VARIANCE	4.5E-07	3.9E-07	T STAT	-25.13668
OBSERVATIONS	50	50		

- The scores are lower than those produced by the champion itself (which followed the optimal path). This is probably due to the randomly generated weights during each step. But, we can also see that the standard deviation of the experiments is low, thus there is performance consistency in the results produced by PRL.
- The score produced by PRL is better than that produced by DDL. With an average maximum of 99.26% compared to 98.93% of DDL, the difference is statistically significant ($p < 0.001$ - Table 6) and the distribution is well separated (Figure 4). Similarly, comparing both maximums of 99.40% (PRL) to 99.05% (DDL), we see a statistically significant difference. Giving DDL more time to train (60 epochs) does not improve its performance (early stop almost always occurs before the epoch number).
- The standard deviation of PRL is lower (better) than that of DDL (Table 6). We believe that this confirms that PRL is a more robust approach and more resilient to random weight initialization.
- During the PRL, the Shannon value consistently decreased at every step (Table 5) of the process. This can be seen as an increasing organization/informational density of the model while the model's complexity increases at each step.
- The Shannon entropy of the PRL-based model is lower (better) than that of the DDL-based model; 8.81 *versus* 9.16.

The last two results reinforce the hypothesis that PRL alleviates the VGE. Though more tests must be conducted to further analyze the approach, this preliminary work shows a promising path. Table 7 shows that when using DDL, the Shannon entropy of the last layers in the model is lower than that of those in the PRL-trained model (bold values for lowest entropy in Table 7). Calculated entropy for each layer of champion 1 are displayed for comparison.

Table 7. Comparison of Shannon entropy between layers.

LAYER #	TYPE	DDL	PRL	Ref. Champion
2	CONV2D	9.162	**8.815**	
3	SEPCNV2D	8.372	**8.234**	8.235
4	CONV2D	15.159	**15.138**	15.142
5	DENSE	14.776	**14.727**	*14.715*
6	DENSE	**11.683**	11.687	11.691
7	CONV2D	14.155	**14.081**	*14.054*
8	CONV2D	14.186	**14.077**	*14.068*
9	DENSE	12.104	**12.101**	*12.095*
10	DENSE	**11.684**	11.688	11.689

11	DENSE	**17.405**	17.512	17.517
12	DENSE	**11.689**	11.711	11.713
13	DENSE	**12.466**	12.588	*12.582*

This hints that the standard training (DDL) is primarily affecting the last layers within the model due to the VGE. The DDL model stores its information in those layers more densely, while in PRL, the weight adjustment and information storage are more evenly distributed. The total Shannon entropy is lower in PRL than in DDL.

Table 8. DDL *vs.* PRL comparison at every evolutionary/complexification step.

STEP	DDL Max. score	PRL Mean score	STEP	DDL Max. score	PRL Mean score
0	92.140	**92.144**	10	98.760	**99.161**
1	**98.160**	97.679	11	98.870	**99.173**
2	98.130	**98.347**	12	98.870	**99.179**
3	98.470	**98.879**	13	98.760	**99.193**
4	98.430	**99.007**	14	98.860	**99.186**
5	98.420	**99.048**	15	98.750	**99.192**
6	98.560	**99.115**	16	98.860	**99.239**
7	98.780	**99.105**	17	98.860	**99.239**
8	98.770	**99.135**	18	98.820	**99.243**
9	98.940	**99.138**	19	98.790	**99.258**

Table 8 shows that if at each step we train the same model (resetting its weights first) using DDL, it both achieves lower final accuracy (performs worse) and based on its Shannon entropy score, stores less information. The performance differences between DDL and PRL trained models increases as they become more complex and grow deeper. The Figure 4. DDL and PRL score distribution on Figure 5. Visual graph of experiment 1 shows a visual graph of the results.

3 experimental results are displayed:

- Evolution score and Shannon retrieved during phase 1 of champion 1 creation.
- Mean DDL score and Shannon at each evolutionary step of champion 1 history.
- Mean PRL score and Shannon of the champion model growth by mutation step.

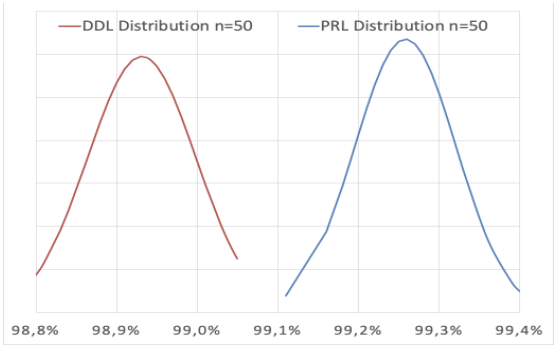


Figure 4. DDL and PRL score distribution on the MNIST dataset.

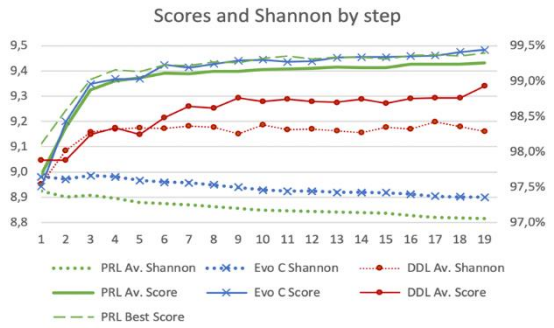


Figure 5. Visual graph of experiment 1 results (MNIST).

Plain lines represent the score, dotted lines represent Shannon entropy and for comparison, the PRL best score is shown as a dashed line. We see that the Shannon score at each step when using DDL of the current step topology is higher (worse) than that of the PRL-based model. Generalization tests (later in this paper) shows that this behavior is reproducible. Further tests must be conducted to conclude whether this behavior applies to any other complex models if we were to build a phylogenetic path and apply the PRL method. Alternatively, perhaps an artificial PRL approach could be used, where any deep model is re-built up one layer at a time and retrained at every step using either an artificially created output layer (of the correct output layer length) until the last layer [29] or by re-attaching the last layer to each consecutive layer and then re-training the model. These artificial approaches of building a path are limited to simple and mostly sequential topologies. Such limitations are not present when it comes to neuroevolution based path building.

5. DISCUSSION

5.1 Reproducibility

This sub-section attempts to answer the following questions: 1/Are these results reproducible with another complex model? 2/ What is the condition of reproducibility? If that condition is the model's complexity, how is such complexity defined?

5.1.1 Reproducibility of PRL Results

To answer the questions, we redo the whole experiment again. For the purpose of reproducibility, we now use another framework, PlaidML and another seed model to generate a new champion. For control, we used the same dataset, the same DDL rules and the same PRL method. The initial model test 2 (Table 9) used in this experiment is narrower, but deeper, as compared to the one in the previous experiment.

Table 9. Initial model test 2.

Layer #	Type	Output	Params
1	InputLayer	N, 28, 28, 1	0
2	Conv2D	N, 27, 27, 6	30
3	MaxPooling2D	N, 9, 9, 6	0
4	Flatten	N, 486	0
5	Dense	N, 10	4870

Table 10 shows the metrics of champion 2 generated from the initial model test 2 (Table 9) during neuroevolution phase 1 of the method.

Table 10. Champion 2 results (MNIST).

Score	Cycle	Generation	Parameters	Nodes	Layers
0.9943	144	28	226 592	39	14

Champion 2 topology generated is smaller, but with a more complex structure, than champion 1 generated in the first experiment. Champion 2 has been generated with 28 evolutionary steps. Furthermore, champion 2 is much harder to train than "initial model 2". Champion 2 epoch time is 15 times that of "initial model 2". Applying DDL to champion 2 gives the following results (MNIST):

DDL average score: 98.90% +/- 0.001 (n=16)
DDL maximum score: 99.08%.

In comparison to the baseline result of the generated champion 2 using neuroevolution, the score we get using DDL with champion 2 topology is lower 99.08% at max. *versus* 99.43% (Table 10). In Table 11, we see that PRL is still more efficient than the DDL approach. The original score of the champion is on average better, which is consistent with our earlier experiments.

Table 11. Results of PRL, DDL applied to champion model 2 (MNIST).

STEP	STD. DEV.	PRL Av. SCORE	DDL Av. SCORE	CHAMP. 2 SCORE	STEP	STD. DEV.	PRL Av SCORE	DDL Av. SCORE	CHAMP. 2 SCORE
0	0.72%	94.31%	**96.37%**	94.60%					
1	0.59%	95.81%	**96.09%**	95.96%	15	0.05%	**99.11%**	98.71%	99.14%
2	0.48%	96.48%	**97.38%**	95.35%	16	0.07%	**99.11%**	98.96%	99.18%
3	0.26%	**97.78%**	97.33%	97.72%	17	0.05%	**99.15%**	98.96%	99.09%
4	0.12%	98.28%	**98.46%**	98.35%	18	0.06%	**99.19%**	98.84%	99.15%
5	0.13%	**98.54%**	98.47%	98.34%	19	0.05%	**99.17%**	98.98%	99.20%
6	0.09%	**98.73%**	98.59%	98.71%	20	0.06%	**99.18%**	98.93%	99.24%
7	0.11%	98.50%	**98.75%**	98.40%	21	0.06%	**99.18%**	98.97%	99.31%
8	0.11%	98.56%	**98.63%**	98.60%	22	0.04%	**99.14%**	98.91%	99.31%
9	0.20%	98.51%	**98.69%**	98.73%	23	0.06%	**99.14%**	98.90%	99.24%
10	0.11%	98.73%	**98.80%**	98.84%	24	0.07%	**99.14%**	99.02%	99.32%
11	0.22%	98.67%	**98.87%**	98.84%	25	0.07%	**99.18%**	98.86%	99.33%

12	0.11%	**98.91%**	98.71%	98.99%	26	0.08%	**99.13%**	98.97%	99.37%
13	0.11%	**98.98%**	98.86%	99.04%	27	0.06%	**99.18%**	98.92%	99.35%
14	0.06%	**99.01%**	98.92%	99.07%	28	0.06%	**99.19%**	98.90%	99.43%

The important result of that experiment is that the initial steps with simpler topology where the VGE is not important had higher scores when using DDL than when using PRL. From step 12 onward, the accuracy/performance achieved by PRL is higher, even though the model was more complex.

5.1.2 Complexity of Model Criteria

When referring to model complexity, we assume that the model has a lot of branches, is deep and is non-sequential. We believe that the more complex (in terms of topology) a model is, the more beneficial it would be to train it using PRL. Thus, in order to further explore these assumptions, we conducted another experiment where we changed the neuroevolutionary phase 1 selection rules.
In this third experiment, we added a rule to the selection process to put more weight on selecting those models which trained the quickest (model training speed was weighted into the final fitness score). With this approach, a model with the same accuracy as another, but with a shorter learning speed (aka epoch time), is selected to enter the HOF. This selection pressure resulted in our system generating champions that are quick to train and less complex, therefore less sensible to vanishing-gradient effect.

Champion 3 generated (MNIST):
Score: 99.40% Shannon: 8.4799

DDL average results for champion 3 model: | PRL average results for champion 3 model:
Score: 99.37% Shannon: 8.4150 | Score: 99.33% Shannon: 8.3535

In this experiment, Shannon value is still lower when using PRL as compared to DDL. But, the difference in the results of this experiment are less drastic.

The PRL complexity definition is therefore not only the topological complexity (total parameters, total nodes and node links), but is also linked to the learning efficiency (amount of time it takes to learn) of the model. The more difficult it is for the model to learn a dataset, the more complex its structure needs to be and the more effect PRL method will have on its training.

5.1.3 Experiment with a Larger Dataset (TinyImageNet)

In this fourth experiment, we used the TinyImageNet dataset with 200 classes and relatively small number of training samples (more difficult to learn) and for reproducibility, no data augmentation. 50 tests were run using DDL and PRL.

Champion 4 generated (TinyImageNet):
19,771,676 parameters, 65 nodes, 30 layers, 44 generations
Score: 42.87% Shannon: 9.3795

DDL average results for champion 3 model: | PRL average results for champion 3 model:
Score : 38.37% Shannon : 9.3833 | Score : 41.79% Shannon : 9.3611

PRL training again produces a better result than DDL.
It should be noted that no data augmentation or extra pre-processing was applied to the dataset. Data augmentation is a common approach with this dataset due to the few samples it contains for each class. Given that our goal is to compare "apples to apples" and find the relative performance of one method compared to another, we applied both PRL and DDL to the original pure TinyImageNet dataset.

5.2 Transferability

In this sub-section, we try to answer the following two questions: 1. Can we use PRL to retrain the model from previous experiments on new datasets from the same problem domain? and 2. Can PRL allow a model to generalize from one dataset to another in the same problem domain better than DDL? In order to answer these questions, we applied PRL to seed and phylogenetic path of champion 1 again, but trained it on a different dataset. The purpose of this experiment is to evaluate whether a model with its recorded evolutionary path from one dataset can be applied on another, but related, dataset.

5.2.1 Experiment 5: FASHION MNIST Dataset Champion 1

The two learning methods are re-applied to the FASHION MNIST dataset. This dataset has the same input and output shape as the standard MNIST. In this dataset, the classification is done on various fashion objects (dresses, shoes, …ext.) rather than digits. This dataset is found to be more complex than the standard MNIST.

DDL mean score for champion 1 - FashionM | PRL mean score for champion 1 - FashionM
Score: 90.44% Shannon: 9.0238 | Score: 91.98% Shannon: 8.4813

This experiment shows that we can re-apply PRL to an existing model and re-train it on a related, but different, dataset. PRL provides a better result than DDL. For comparison, the SOTA convolutional NN applied to the FASHION MNIST is 91.4% without data augmentation [30]. Our 91.98% is a competitive result that outperforms the SOTA, even though the model trained by PRL was not evolved for that specific dataset. This is an interesting result. One potential implication of this result is that PRL might allow us to more easily re-train existing model architectures on new, but related, problem domains for which they were not originally designed and still achieve very high performance.

Table 12. Shannon layer comparison for FASHION MNIST.

LAYER	TYPE	DDL	PRL
2	CONV2D	9.0238	**8.4813**
3	SEPCNV2D	8.307	**8.0439**
4	CONV2D	15.1492	**15.1338**
5	DENSE	14.7804	**14.7331**
6	DENSE	11.6941	**11.6841**
7	CONV2D	14.1506	**13.9888**
8	CONV2D	14.1568	**13.9923**
9	DENSE	12.1115	**12.1017**
10	DENSE	11.6922	**11.6905**
11	DENSE	**17.3703**	17.4783
12	DENSE	**11.6984**	11.71
13	DENSE	**12.3777**	12.5757

Table 12 shows again that PRL is better able to alleviate the VGE. The first layers have better Shannon entropy values when a model is trained through PRL and the last layers have better entropy values when DDL is used to train the model.

5.2.2 Experiment 6: CIFAR10 Gray

In this experiment, we train the model on the CIFAR10 dataset converted into grayscale (C10G). For this experiment, we also scaled it to the 28*28*1 resolution, then gray-scaled it, so that we can use the same model again and test its transferability. This downgraded dataset is much more difficult to train than the MNIST.

DDL mean results for champion 1 - C10G: | PRL mean results for champion 1 - C10G
Score: 54.40 Shannon: 9.1690 | Score: 65.01 Shannon: 8.9345

These results again show the PRL's ability to generalize and retrain an existing model on a new, but related, dataset (with the same shape). In our experiments, PRL is consistently producing better results than DDL, both in terms of accuracy and information density (Shannon entropy values).

5.3 Learning Time

While neuroevolution algorithms are known to need more time to achieve good results (though still significantly less when compared to the time needed by experts to design similar problem specific topologies manually), the PRL adds an alternative to the DDL methods.

Overall, the PRL process requires more steps (19 steps*20 epochs at most in experiment 1), but the Epochs' durations are shorter during the early steps (3 to 15 times less). DDL required at most 60 epochs. Further experiments should be conducted to optimize learning time. We believe that methods like layer-freezing and optimizing the number of epochs will help in this. We conducted preliminary PRL tests

with 4 epochs for half the steps followed by an increase in number of epochs until the last step, where the results were still superior as compared to DDL, while decreasing the needed time.

DDL learning time:	60 epoch*x (x = champion epoch duration in seconds)	= 60.0x
PRL in 1st experiment:	(20 epoch*19 step)*(x/3)	= 126.6x
PRL in 1st experiment with step epoch number optimized:		
	(4 epoch*10 step+12 epoch*9 step)*(x/3)	= 49.3x

PRL takes more time than DDL, but with a simple optimization, the runtime could be reduced below the DDL time.

6. CONCLUSION

Based on our experiments and results, PRL has consistently outperformed DDL, primarily by alleviating the VGE problem. We believe that this is the way in which it functions due to the Shannon entropy values calculated for each layer. These values are lower in deeper layers in the models trained by PRL than those trained by DDL. Furthermore, PRL is more resilient to random-weight initialization as compared to DDL. We re-ran the PRL experiment on the same seed model and with the same phylogenetic path, but with each seed model having randomly generated initial synaptic weights. We found that the performances of the evolved champion models were all very similar and more consistent than when randomly initializing a full model and training it with DDL. Our experiments on transferability also show that the method is effective in retraining models on related datasets. This potentially opens the door to further research into the method's use in transfer learning, where a model with its phylogenetic path can be effectively retrained on another dataset or an updated version of the same dataset. We believe that further research is needed into this domain. The combination of neuroevolution, where model/architecture evolution is synergized with training, will yield better performing systems, as compared to systems where the model is trained all at once (DDL). We think that this method might be particularly effective in training very deep and very complex models, where DDL might struggle. Our future work will concentrate on further expanding and exploring this method.

REFERENCES

[1] D. Silver, David et al., "Mastering the Game of Go with Deep Neural Networks and Tree Search," Nature, vol. 529, pp. 484-489, DOI: 10.1038/nature16961, 2016.

[2] P. Vikhar, "Evolutionary Algorithms: A Critical Review and Its Future Prospects," Proc. of the IEEE Int. Conf. on Global Trends in Signal Process., Inf. Comp. and Comm. pp. 261-265, Jalgaon, India, 2016.

[3] F. Gomez, J. Schmidhuber and R. Miikkulainen, "Accelerated Neural Evolution through Cooperatively Coevolved Synapses," Journal of Machine Learning Research, vol. 9, pp. 937-965, 2008.

[4] R. De Nardi, J. Togelius, O. Holland and S. Lucas, "Evolution of Neural Networks for Helicopter Control: Why Modularity Matters," Proc. of the IEEE Int. Conf. on Evolutionary Computation, pp. 1799-1806, DOI: 10.1109/CEC.2006.1688525, Vancouver, Canada, 2006.

[5] V. Heidrich-Meisner, C. Igel, B. Hoeffding and Bernstein, "Races for Selecting Policies in Evolutionary Direct Policy Search," Proc. of the 26th Annual Int. Conf. on Machine Learning (ICML '09), vol. 51, DOI: 10.1145/1553374.1553426, 2009.

[6] J. Lehman et al., "The Surprising Creativity of Digital Evolution: A Collection of Anecdotes from the Evolutionary Computation and Artificial Life Research Communities," Massachusetts Institute of Technology, Artificial Life, vol. 26, no. 2, pp. 274–306, 2020.

[7] F. Such et al., "Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning," arXiv, DOI: 10.48550/arXiv.1712.06567, 2017.

[8] X. Zhang, J. Clune and K. Stanley, "On the Relationship between the OpenAI Evolution Strategy and Stochastic Gradient Descent," arXiv: 1712.06564, DOI: 10.48550/arXiv.1712.06564, 2017.

[9] J. Lehman, J. Chen, J. Clune and K. Stanley, "ES Is More Than Just a Traditional Finite-difference Approximator," Proc. of the Genetic and Evolutionary Computation Conference (GECCO '18), pp. 450-457, DOI: 10.1145/3205455.3205474, 2018.

[10] E. Conti, Edoardo et al., "Improving Exploration in Evolution Strategies for Deep Reinforcement Learning *via* a Population of Novelty-seeking Agents," Proc. of the 32nd Int. Conf. on Neural Information Processing Systems (NIPS'18), pp. 5032–5043, 2017.

[11] J. Metzen, M. Edgington, Y. Kassahun and F. Kirchner, "Performance Evaluation of EANT in the Robocup Keepaway Benchmark," Proc. of the 6th Int. Conf. on Machine Learning and Applications (ICMLA 2007), pp. 342-347, DOI: 10.1109/ICMLA.2007.23, 2008.

[12] F. Gomez, J. Schmidhuber and R. Miikkulainen, "Accelerated Neural Evolution through Cooperatively

Coevolved Synapses," JMLR, vol. 9, pp. 937-965, DOI: 10.1145/1390681.1390712, 2008.

[13] K. Stanley and R. Miikkulainen, "Evolving Neural Networks through Augmenting Topologies," Evolutionary Computation, vol. 10, pp. 99-127, DOI: 10.1162/106365602320169811, 2002.

[14] E. Real, A. Aggarwal, Y. Huang and Q. Le, "Regularized Evolution for Image Classifier Architecture Search," Proc. of AAAI Conf. on Artificial Intellig., vol. 33, DOI: 10.1609/aaai.v33i01.33014780, 2018.

[15] A. Gaier and D. Ha, "Weight Agnostic Neural Networks," arXiv: 1906.04358, DOI: 10.13140/RG.2.2.16025.88169, 2019.

[16] S. Hochreiter, Untersuchungen zu dynamischen neuronalen Netzen, Diploma Thesis, Josef Hochreiter Institut fur Informatik, Technische Universitat Munchen, Germany, 1991.

[17] F. Informatik, Y. Bengio, P. Frasconi and J. Schmidhuber Jfirgen, "Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies," Chapter of Book: A Field Guide to Dynamical Recurrent Neural Networks, pp. 237 – 243, DOI: 10.1109/9780470544037.ch14, IEEE Press, 2003.

[18] Y. Bengio, P. Simard and P. Frasconi, "Learning Long-term Dependencies with Gradient Descent Is Difficult," IEEE Transactions on Neural Networks, vol. 5, pp. 157-166, DOI: 10.1109/72.279181, 1994.

[19] R. Pascanu, T. Mikolov and Y. Bengio, "On the Difficulty of Training Recurrent Neural Networks," Proc. of the 30th Int. Conf. on Machine Learning, JMLR: W&CP, vol. 28, Atlanta, Georgia, USA, 2013.

[20] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," Proc. of the IEEE Conf. on Comp. Vision and Pattern Recog. (CVPR), pp. 770-778, DOI: 10.1109 CVPR.2016.90, 2016.

[21] X. Glorot, A. Bordes and Y. Bengio, "Deep Sparse Rectifier Neural Networks," Proc. of the 14th Int. Conf. on Artificial Intelligence and Statistics, vol. 15, pp. 315-323, Fort Lauderdale, FL, USA, 2011.

[22] Y. Lecun, L. Bottou, G. Orr and K.-R. Müller, "Efficient BackProp," Chapter in Book: Neural Networks: Tricks of the Trade, vol. 7700, pp. 9-48, DOI: 10.1007\/3-540-49430-8_2, 1998.

[23] X. Glorot and Y. Bengio, "Understanding the Difficulty of Training Deep Feedforward Neural Networks," Journal of Machine Learning Research, vol. 9, pp. 249-256, 2010.

[24] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," arXiv: 1502.03167, DOI: 10.48550/arXiv.1502.03167, 2015.

[25] Y. Lecun et al., "Backpropagation Applied to Handwritten Zip Code Recognition," Neural Computation, vol. 1, pp. 541-551, DOI: 10.1162 neco.1989.1.4.541, 1989.

[26] H. Noh, T. You, J. Mun and B. Han, "Regularizing Deep Neural Networks by Noise: Its Interpretation and Optimization," Proc. of the 31st Conf. on Neural Inf. Process. Sys. (NIPS), Long Beach, USA, 2017.

[27] S. Enrique, J. Hare and M. Niranjan, "Deep Cascade Learning," IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 11, pp. 5475 – 5485, DOI: 10.1109/TNNLS.2018.2805098, 2018.

[28] C. Shannon and W. Weaver, The Mathematical Theory of Communication, Note 78, p. 44, 1963.

[29] J. Schmidhuber, "Learning Complex, Extended Sequences Using the Principle of History Compression," Neural Computation, vol. 4, pp. 234-242, DOI: 10.1162/neco.1992.4.2.234, 1992.

[30] O. Granmo et al., "The Convolutional Tsetlin Machine," arXiv: 1905.09688v5, DOI: 10.48550/arXiv.1905.09688, 2019.

ملخص البحث:

رغم التّطور الكبير الذي طرأ على تدريب الشبكات العصبية، إلّا أنّه ولسوء الحظّ، فإنّ نقطة القوة في الشبكات العصبية العميقة – وهي عمقها - تشكل مشكلةً وهي تضاؤل الميْل؛ وذلك لصعوبة التّدريب المعمَّق للطّبقات الأعمق.

تقترح هذه الورقة طريقة "تعلُّم الإعادة المكتسب خلال التّطوّر النوعي"، وهي طريقة تعلُّم تتغلّب بشكلٍ جوهري على مشكلة تضاؤل الميْل. وعلى العكس من طرق التعلُّم بالبواقي، فإنّ الطريقة المقترحة لا تُقيّد بِنية النّموذج. وبدلاً من ذلك، فهي تقوّي عناصر من التّطوُر العصبي، وتعلُّم النّقل، والتّدريب طبقةً طبقةً. ونبين في هذه الورقة أنّ الطريقة المقترحة الجديدة قادرة على إنتاج نموذجٍ ذي أداءٍ أفضل. وبحساب إنتروبيا شانون للأوزان، نبين أنّ الطبقات الأعمق يتمّ تدريبها على نحوٍ أكثر تفصيلاً بحيث تحتوي على معلومات أكثر بدلالةٍ إحصائية مقارنةً بالحالات التي يتمّ فيها تدريب النّموذج بالطّرق التّقليدية.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 08, No. 03, September 2022.

AN ENHANCED APPROACH FOR CP-ABE WITH PROXY RE-ENCRYPTION IN IOT PARADIGM

Nishant Doshi

(Received: 1-Feb.-2022, Revised: 30-Mar.-2022, Accepted: 27-Apr.-2022)

ABSTRACT

In Internet of Things (IoT), encryption is a technique in which plaintext is converted to ciphertext into make it non-recovered by the attacker without secret key. Ciphertext policy attribute-based encryption (CP-ABE) is an encryption technique aimed at multicasting feature; i.e., user can only decrypt the message if the policy of attributes mentioned in the ciphertext is satisfied by the user's secret key attributes. In literature, the authors have improvised the existing technique to enhance the naïve CP-ABE scheme. Recently, in 2021, Wang et al. have proposed the CP-ABE scheme with proxy re-encryption and claimed it to be efficient as compared to its predecessors. However, it follows the variable-length ciphertext in which the size of ciphertext is increased with the number of attributes. Also, it leads to computation overhead on the receiver during decryption which will be performed by the IoT devices. Thus, in this paper, we have proposed on improved scheme to provide the constant-length ciphertext with proxy re-encryption to reduce the computation and communication time. The proposed scheme is secured under Decisional Bilinear Diffie-Hellman (DBDH) problem.

KEYWORDS

Attribute, Multi-authority, Proxy re-encryption, Constant length.

1. INTRODUCTION

In this IoT era, encryption techniques are playing vital role to achieve security and confidentiality. In a conventional symmetric key-based encryption scheme, sender and receiver possess the same key for communication. So, if one of the users compromised, then the entire scheme is compromised. To resolve this problem, [1] proposed the public key or asymmetric key encryption scheme, in which the receiver gives his/her public key to the sender for the encryption of the message, so only the receiver is able to decrypt the message. But, this scheme does not support the efficient multicast because a multicast sender has to encrypt the message a number of times equal to the number of receivers. The other problem is that the sender requires to remember the public key of the receiver. To overcome this problem, in [2] authors propose the Identity-based Encryption (IBE) in which the sender encrypts the message based on the receiver's unique id, like SSN, email id, …etc. But, this scheme also does not support multicast; so in [3], the authors propose the Fuzzy IBE system in which user with id X can only be able to decrypt the ciphertext entitled for X' if and only if $|X - X'| > y$, where y is the initial threshold value.

In research, the idea of IBE is generalized to solve the computation overhead during multicast and is called Attribute-based Encryption (ABE). ABE is classified into two variants; i.e., Key Policy Attribute-based Encryption (KP-ABE) [4] and Ciphertext Policy Attribute-based Encryption (CP-ABE) [5]. As the names suggest, in KP-ABE, policy of attributes is attached with secret key, whereas in CP-ABE, policy of attributes is attached with ciphertext. Indeed, CP-ABE gives more control to the sender in terms of selecting the intended recipients. In this research, we are focusing on the CP-ABE. In [5], the authors have proposed the single authority-based approach in which authority will generate the entire secret key of users.

As the existing approaches deal with single authority, they suffer from issues *viz.* (i) key escrow: authority can regenerate the secret key on behalf of any user (ii) computation overhead on the authority to generate the entire secret key of all system users. To deal with this issue, in [6]–[10], the authors have proposed various approaches based on multi-authority systems. In IoT, this will be helpful to design the decentralized approaches based infrastructure.

All the approaches mentioned so far requires sender to re-encrypt the same message for different policies. This leads to computation overhead which can be mitigated by proxy-based cloud systems. In proxy re-encryption, the proxy will re-encrypt without any knowledge of the secret key of the user. In

N. Doshi is with School of Technology, PANDIT Deendayal Energy Uni., Raisan, India. Email: nishant.doshi@sot.pdpu.ac.in

[11]–[15], the authors have proposed various approaches to deal with proxy-based re-encryption mechanism. In IoT, this will be helpful to reduce the computation overhead on IoT devices.

All the approaches mentioned so far deal with variable-length ciphertext; i.e., length of the ciphertext increases with the number of attributes. This will lead to communication overhead as well as computation overhead on the receiver side. In IoT paradigm, we required energy efficient-approaches as to run on the deployed sensors in the field. To deal with this issue, in [16]–[19], the authors have proposed approaches based on constant-length ciphertext. More details on these approaches are given in the next section.

1.1 Our Contribution

Amidst of the above concerns, research will lead to the need of one system having all features. On the other side, we have schemes to give either of the features as reported in literature [20]–[29]. Thus, in this paper, we propose the collusion-resistant CP-ABE scheme which provides the proxy re-encryption to make our scheme applicable in the scenario where compromised users' leaked decryption keys can be traced and nullified. Our scheme works for the threshold case; i.e., the attributes in the ciphertext must be equal to the subset of user's attributes in his/her secret key. We proposed new protocol to address this problem and show the efficiency compare to the existing protocols. The security of this protocol is based on DBDH assumptions as their predecessors.

1.2 Paper Organization

The rest of the paper is organized as follows. Section 2 deals with a literature review in this field. Section 3 deals with the hardness problems used for the security of the proposed work. Section 4 showcases the proposed work. Section 5 deals with the security and computation analysis. The conclusion and references are presented at the end.

2. LITERATURE REVIEW

In this section, we conduct a literature survey on the various approaches in CP-ABE.

2.1 Multi-authority

The original CP-ABE scheme [5] is dealing with single-authority environment. A single-authority system requires the entire trust on the same authority, so if authority-compromised or behaves maliciously, then the entire system will be compromised. In addition, it deals with computation overhead on authority as to generate the entire secret keys of all system users. To overcome these issues, in [6], the authors firstly propose the idea of multi-authority systems. In a multi-authority system, there is one central authority (CA) and multiple attribute authorities (AAs). As we observed, this scheme requires mutual trust between AAs and the CA must be present to manage the attribute authorities and add new AAs. The CA is able to decrypt any ciphertext, which can harm the system. In [7]–[10], [30]-[31], the authors proposed different approaches to deal with the multi-authority system.

2.2 Proxy Re-encryption (PRE)

It's a technique in which an untrusted proxy server will translate a ciphertext encrypted under Alice's public key to a ciphertext encrypted under bob's public key. This can be useful in email forwarding applications. For PRE, Alice can generate a PRE key, which she can give to proxy, so there is no need to store it at the user side. Proxy can get no information regarding secret of Alice from PRE key. Upon incoming ciphertext, proxy can apply the PRE key to get the required ciphertext. In [32], the authors introduced the notion of PRE. In [33], the authors proposed the bidirectional PRE scheme. In [34], the authors proposed the first unidirectional PRE scheme. In [35], the authors proposed the IBE-PRE scheme which converts ciphertext encrypted under Alice's identity to one encrypted under bob's identity. Their scheme is secure under random Oracle. In [36], the authors proposed the IBE-PRE in standard model. In [37], the authors proposed the first AB-PRE scheme which is bidirectional and based on key policy scheme. In [38], the authors proposed the first CP-ABE-PRE scheme. In [39], the authors proposed the variable-length CP-ABE-PRE scheme. In [40], the authors proposed the constant

ciphertext length for CP-ABE-PRE scheme, but they required the same number of attributes in policy as in secret key. In [11]–[15], the authors have proposed various approaches for improving the existing schemes.

2.3 Ciphertext Length

All the approaches mentioned so far deal with the variable-length ciphertext approach; i.e., length of the ciphertext increases with the number of attributes. This will increase the computation overhead on the receiver due to access amount of operations during decryption. In [41], the authors firstly introduced the concept of constant-length ciphertext using the (t, t) threshold system. As mentioned, it requires the same set of attributes in ciphertext as well as in secret key for successful decryption. This makes the scheme of [41] usable in limited scenarios. In [42], the authors proposed the constant-length ciphertext in threshold ABE based on the dynamic threshold encryption scheme from [43]. In [16]–[19], the authors have proposed various schemes to improve constant-length ciphertext.

Based on our literature survey, we have schemes available for the multi-authority or constant-length ciphertext. However, none of the approach available in research provides all features in a single scheme. In addition, to use a different scheme for each feature can be an overhead on the system users. Thus, in this paper, we have proposed a single scheme to provide all these features.

3. PRELIMINARIES

In this section, we present the preliminaries as well as the hardness problems that will be utilized throughout the paper.

3.1 Bilinear Group

The security of the proposed system is based on the algebraic group called the bilinear groups based on a bilinear map. As we are using bi-linear map function for pairing operations, we have taken Decisional Bilinear Diffie Hellman hardness problem.

Definition 1 (Bilinear map). Consider cyclic multiplicative group G_1, G_2 and G_3 of prime order p and generators g_1, g_2 and g_3, respectively, as well as a deterministic bilinear map function e: $G_1 \times G_2 \rightarrow G_3$ with the following requirements.

- Bi-linearity : For all $x \in G_1, y \in G_2, a, b \in Z_p$, $e\ (x^a, y^b) = e(x, y)^{ab}$.
- Non-degeneracy: $e\ (g_1, g_2) \neq 1$.
- Efficiency: e must be a time-efficient function.

Definition 2 (Discrete Logarithm Problem (DLP)). Find an integer $x \in Z_p$, such that $h = g^x$ whenever such integer exists given two group elements g and h.

Definition 3 (Decisional Bilinear Diffie Hellman (DBDH) Problem). In prime-order group G with generator g, on input $g, g^a, g^b, g^c \in G$, check whether $c = ab$ or not.

3.2 Proposed Construction

The proposed scheme consists of a number of polynomial algorithms as follows.

- **Setup**: It runs by central authority (CA) to generate the private and public parameters of the system.
- $\boldsymbol{AA_i}$ **Setup**: It runs by the respective Attribute Authority (AA) to generate the parameters of authority.
- **KeyGen**: It runs by the CA to generate part of the secret keys of the users. It consists of the following-sub algorithms.
    - **RKGen:** It runs by the user to generate re-encryption key for the proxy servers.
- **RequestAttributeSK:** It runs by AA to give the secret component respective to the attribute in the user's secret key.
- **Encrypt**: It runs by the sender to convert the plaintext into ciphertext based on the access policy. It consists of the following sub-algorithms.
    - **ReEncrypt**: It runs by the proxy server to convert ciphertext from one policy to another policy.

- **Decrypt**: It runs by the receiver to get the plaintext from the ciphertext if the access policy is satisfied; else a random message will be given.

4. THE PROPOSED SCHEME

The proposed scheme(s) consists of the following polynomial algorithms. The schematic diagram of the proposed scheme is given in Figure 1.

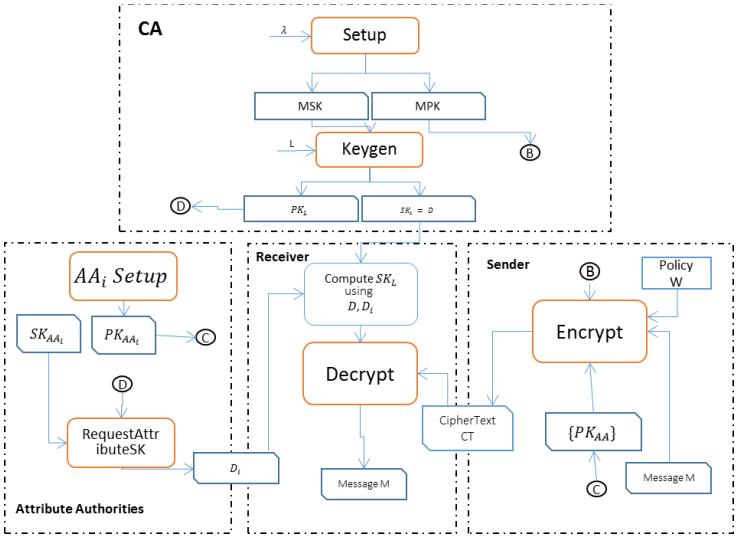


Figure 1. Schematic diagram for the proposed scheme.

Set-up: It executes by the CA as follows.

- Selects a bilinear group G_0 of prime order p with generator g.
- Selects exponents $y, \beta \in_R Z_p$.
- Computes $h = g^{\beta}, Y = e(g,g)^{y}$.
- Computes Master Public Key $MPK = G_0, g, h, Y$.
- Computes Master Secret Key $MSK = (\beta, y)$.

AA_i setup: It runs by AA_i to create parameters for attribute *i*.

- Chooses exponent $\alpha_i \in_R Z_p$.
- Computes Public Parameter $PK_i = g^{\alpha_i}$.
- Computes Secret Parameter $SK_i = \alpha_i$.

Keygen (MSK,u) : It runs by the CA to create the secret key (SK) for user *u*. L denotes the attributes' list. It is exemplified from this algorithm that CA is responsible for the generic parameters, not the attributes of the users. Thus, compromising the CA cannot compromise the system; i.e., the system is secure against key escrow. Also, due to the unique r value in the user's secret key, the proposed scheme is secure against collusion attack.

- Chooses $r \in_R Z_p$.
- Computes secret key $SK_u = g^{(y+r)/\beta}$.
- Computes public key $PK_u = g^r$.
- Sets attribute list $L_u = \emptyset$.

RKGen(MPK,W,W',SK_L) : This algorithm runs by user *u*, consisting of attribute set $AS \subseteq L$ and satisfying access policy *W*. This algorithm gives proxy re-encryption key which can be used to convert ciphertext with access policy *W* into ciphertext with access policy *W'*. Here, $n = |AS| = |W|$; i.e., number of attributes in access policy W.

- Generates $d, g_1 \in_R Z_p$.
- Computes $C = Encrypt\ (MPK, {g_1}^{nd}, W')$.
- Computes $R = D \prod_{v_{i,j} \in AS} (D_{i,j} {g_1}^{d})$.
- $RK_{AS \to W'} = < C, R, g^r >$

RequestAttributeSK(PK_u,u, L_u): This algorithm runs by AA. AA generates exponent $r_i \in_R Z_p$. H denotes hash function with one-way property.

- Computes $D_i = (g^r)^{\alpha_i}$ and $L_u = L_u + i$.
- Sends D_i and L_u to user.

Encrypt(M,W,PK_1,PK_2,…,PK_N): It runs by the sender by taking the list of attributes for policy as well as message M. It follows the steps below:

- Chooses exponent $s \in_R Z_p$.
- Computes $C_1 = M\, Y^s$.
- Computes $C_2 = g^s$.
- Computes $C_3 = (\prod_{t\in W} PK_i)^s = (\prod_{t\in W} g^{\alpha_i})^s$.
- Computes $C_4 = (h)^s = g^{\beta s}$.
- Final ciphertext $CT = \{C_1, C_2, C_3, C_4, W\}$.

As can be seen from the above steps, we have five components only irrespective of the set of attributes in the final ciphertext. This will achieve the constant-length ciphertext approach.

RKEncrypt(CT_W,$RK_{AS\rightarrow W'}$) : It runs by proxy server to convert the CT_W to $CT_{W'}$. There exists the attribute set $AS \subseteq L$ and it satisfies access policy *W*.

- $C' = \frac{C_1\, e(C_2, g^r)}{e(C_3, R)} = \frac{M}{e(g,g_1)^{n\,s\,d\,\beta}}$
- $CT_{W'} = < C', C, C_3 >$

Decrypt(SK,CT) : It runs by the receiver by taking CT as well as SK as input. It returns M if policy is satisfied; else a random message is given. For simplicity, assume $AS \subseteq L$ and $AS = W$. It is divided in two parts based on CT being the original ciphertext of proxy re-encrypted ciphertext.

If **CT** is an original ciphertext, then the user will follows the below:

$$= \frac{C_1 \cdot e(g^r, C_2) \cdot e(C_3, g^r)}{e\left(C_4, g^{\frac{y+r}{\beta}}\right) \cdot e(C_2, (\prod_{t\in AS} g^{\alpha_t})^r)} = \frac{M \cdot e(g,g)^{y\,s} \cdot e(g,g)^{r\,s} \cdot e(g,g)^{r\,s\,p}}{e(g^s, g^{y+r}) \cdot e(g^s, g^{r\,q})} = \frac{M \cdot e(g,g)^{y\,s} \cdot e(g,g)^{r\,s} \cdot e(g,g)^{r\,s\,p}}{e(g,g)^{y\,s} \cdot e(g,g)^{r\,s} \cdot e(g,g)^{r\,s\,q}} = M.$$

Here, $p = \sum_{t\in W} \alpha_t$ and $q \sum_{t\in AS} \alpha_t$.

If **CT** is a re-encrypted ciphertext, then the user will follow the below:

$$g_1{}^{nd} = Decrypt(SK, CT) = \frac{M\, e(C_3, g_1{}^{nd})}{e(g, g_1)^{n\,s\,d\,\beta}} = M.$$

5. ANALYSIS

As we discussed, ABE has actually evolved from IBE. The security of ABE schemes is also typically modeled on the lines of security of the IBE schemes. The scheme that we propose here is inspired by the one in [4], in which the authors first proposed a scheme for ABE. The scheme is described in a setup that involves a security game amongst an attacker and a challenger, along with a simulator.

The simulator generates an initial parameter and gives it to the challenger. Based on this security game, the ABE schemes can broadly be categorized into two categories *viz. selective secure* and *fully secure*. In *selectively* secure schemes, the attacker announces the target policy ahead of the game, so that the simulator can bind the hardness of the problem with the attributes mentioned in the policy. In *fully* secure schemes, the attacker is not required to announce the target policy initially, as there are sequences of games played between the attacker and the challenger. Figure 2 depicts the security game between the challenger (CA+AAs) and the attacker.
As one can see, the attacker announces the target policy before seeing the public parameters, which makes the proposed scheme a selectively secure model.

5.1 Security Analysis

Theorem 1: The proposed scheme is secure under the DBDH assumption for message indistinguishability.
Proof: Assume that the adversary A gains the advantage ϵ in the security game. Therefore, a simulator

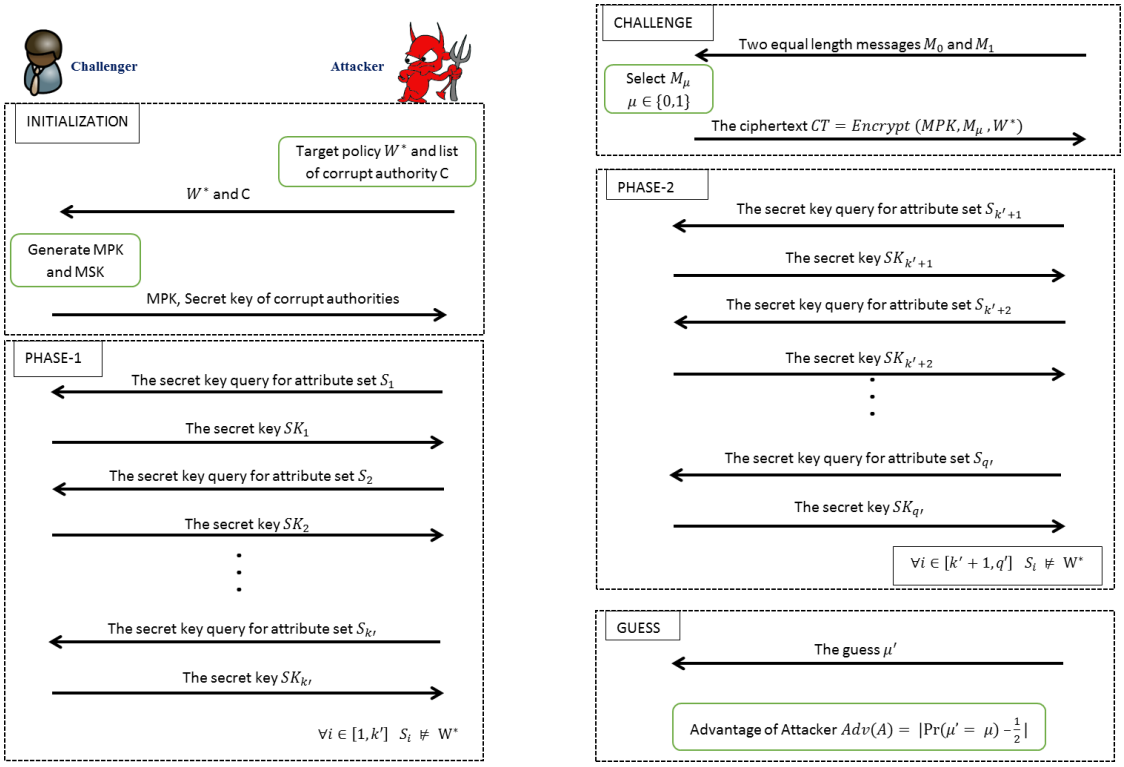


Figure 2. Security game for MA-CP-ABE scheme.

X will be constructed in DBDH assumption with advantage $\frac{\epsilon}{2}(1-\frac{(N')^2}{p})$, where $N' = 2^{n\,n_i}$ represents the number of access structures. The challenger generates $a, b, c, z \in_R Z_p$, $\omega \in_R \{0,1\}$ and g, where g is the generator for group G; so,

$$Z = \begin{cases} e(g,g)^{abc}, & (w=0) \\ e(g,g)^{z}\ , & (w=1) \end{cases}$$

The challenger gives $(g, g^a, g^b, g^c, Z) \in G^4 \times G_1$ to X. Now, A provides target policy $W^* = [W_1^*, W_2^*, \ldots, W_k^*]$ to X. X sets the parameter $Y = e(g^a, g^b) = e(g,g)^{ab}$. For $\alpha'_{i,j}\{i \in [1,n], j \in [1,n_i]\} \in_R Z_p$. X selects secret keys $\alpha_{i,j}\{i \in [1,n], j \in [1,n_i]\}$ and public keys $T_{i,j}\{i \in [1,n], j \in [1,n_i]\}$ as follows:

$$\alpha_{i,j} = \begin{cases} \alpha'_{i,j}, & (v_{i,j} = W_i^*) \\ b\alpha'_{i,j}, & (v_{i,j} \neq W_i^*) \end{cases}$$

$$T_{i,j} = \begin{cases} g^{\alpha'_{i,j}}, & (v_{i,j} = W_i^*) \\ (g^b)^{\alpha'_{i,j}}, & (v_{i,j} \neq W_i^*) \end{cases}$$

X gives $MPK = \left(e, g, h, Y, T_{i,j}\{i \in [1,n], j \in [1,n_i]\}\right)$ to A. X gives the secret parameters of corrupted authorities to A. In **Extract** query L, it requires $v_{i,j} = L_i$ and $v_{i,j} \neq W^*$ because $L \neq W^*$. Thus, X can write $\sum_{v_{i,j} \in L} \alpha_{i,j} = X_1 + bX_2$, where $X_1, X_2 \in Z_p$. Here, X_1 and X_2 showcase the summation of $\alpha'_{i,j}$ values. Thus, X can recalculate X_1 and X_2; it chooses $\beta \in_R Z_p$, calculates $r = \frac{\beta - ua}{X_2}$ and re-computes SK_L as follows:

$SK_L = \{g^{\frac{\beta}{X_2}}(g^a)^{\frac{-u}{X_2}}, \left(g^{ab} g^{\frac{\beta}{X_2}}(g^a)^{\frac{-u}{X_2}}\right)^{1/\beta}, \forall_{v_{i,j} \in L}(g^{\alpha_{i,j}})^{\frac{\beta}{X_2}}((g^a)^{\alpha_{i,j}})^{\frac{-u}{X_2}}\}$.

Therefore, SK_L becomes a correct secret key as follows:

$$\left(g^{ab} g^{\frac{\beta}{X_2}}(g^a)^{\frac{-u}{X_2}}\right)^{1/\beta} = \left((g)^{ab}(g)^{\frac{\beta-ua}{X_2}}\right)^{1/\beta} = g^{(y+r)/\beta}.$$

$g^{\frac{\beta}{X_2}}(g^{a})^{\frac{-u}{X_2}} = g^{\frac{\beta - ua}{X_2}} = g^{r}$ and $(g^{\alpha_{i,j}})^{\frac{\beta}{X_2}}((g^{a})^{\alpha_{i,j}})^{\frac{-u}{X_2}} = (g^{\alpha_{i,j}})^{r} = (T_{i,j})^{r}$.

A will select $AS \subseteq L$ and compute $\prod_{v_{i,j} \in \text{AS}}(T_{i,j})^{r} = g^{r\sum_{v_{i,j} \in \text{AS}} \alpha_{i,j}}$

If $X_2 = 0 \ mod \ p$, then AS ⊆ L with $\sum_{v_{i,j} \in \text{AS}} \alpha_{i,j} = \sum_{v_{i,j} \in \text{W}^*} \alpha_{i,j}$, then X aborts with $\Pr[abort] = \frac{(N')^2}{p}$.

X selects $\mu \in_R \{0,1\}$, calculates $C_1^* = M_\mu Z, C_2^* = g^c, C_3^* = (g^c)^{\sum_{v_{i,j} \in W^*} \alpha_{i,j}}, C_4^* = (g^c)^{\beta}$ and sends $< C_1^*, C_2^*, C_3^*, C_4^*, W^* >$ to A.

Guess: A gives $\mu' \in \{0,1\}$ on μ as guess.

If $\mu' = \mu$, then X sets $\tau' = 0$; else $\tau' = 1$. Appropriate to this, the two cases are as follows:

Case 1: If $\tau = 0$, then $Z = e(g,g)^{abc}$ and ciphertext is valid for M_μ.

$\therefore A$ can output $\mu' = \mu$ with advantage ϵ.

$$\therefore \Pr[\mu' = \mu | \tau = 0 \wedge \overline{abort}] = \frac{1}{2} + \epsilon.$$

$\therefore \Pr[\tau' = \tau | \tau = 0 \wedge \overline{abort}] = \frac{1}{2} + \epsilon$, since X guesses $\tau' = 0$ when $\mu' = \mu$.

Case 2: If $\tau = 1$, the target policy is independent of M_0 and M_1, so that A fails to get μ.

$\therefore A$ can output $\mu' = \mu$ with *NO* knowledge.

$\therefore \Pr[\mu' \neq \mu | \tau = 1 \wedge \overline{abort}] = \frac{1}{2}$.

$\therefore \Pr[\tau' \neq \tau | \tau = 1 \wedge \overline{abort}] = \frac{1}{2}$, since X guesses $\tau' = 1$ when $\mu' \neq \mu$.

From case 1 and case 2, X is having the following advantage in this DBDH game:

$$\Pr[\tau' = \tau] - \frac{1}{2} = \Pr[\tau = 0]\Pr[\tau' = \tau | \tau = 0] + \Pr[\tau = 1]\Pr[\tau' = \tau | \tau = 1] - \frac{1}{2}$$

$$= \frac{1}{2}\Pr[\tau' = \tau | \tau = 0] + \frac{1}{2}\Pr[\tau' = \tau | \tau = 1] - \frac{1}{2} = \frac{1}{2}\{\Pr[\tau' = \tau | \tau = 0] + \Pr[\tau' = \tau | \tau = 1] - 1\}$$

$$= \frac{1}{2}\{\Pr[abort]\Pr[\tau' = \tau | \tau = 0 \wedge abort] + \Pr[\overline{abort}]\Pr[\tau' = \tau | \tau = 0 \wedge \overline{abort}]$$
$$+ \Pr[abort]\Pr[\tau' = \tau | \tau = 1 \wedge abort] + \Pr[\overline{abort}]\Pr[\tau' = \tau | \tau = 1 \wedge \overline{abort}]\}$$

As *"abort"* is not dependent on DBDH challenge, we have:

$\Pr[\tau' = \tau | \tau = 0 \wedge abort] = \Pr[\tau' = \tau | \tau = 1 \wedge abort] = \frac{1}{2}$.

$$= \frac{1}{2}\left\{\frac{(N')^2}{p}\frac{1}{2} + \left(1 - \frac{(N')^2}{p}\right)\left(\frac{1}{2} + \epsilon\right) + \frac{(N')^2}{p}\frac{1}{2} + \left(1 - \frac{(N')^2}{p}\right)\frac{1}{2} - 1\right\}$$

$$= \frac{1}{2}\left\{\left(1 - \frac{(N')^2}{p}\right)\epsilon\right\} = \frac{\epsilon}{2}\left(1 - \frac{(N')^2}{p}\right).$$

5.2 Performance Analysis

In this sub-section, we present the comparative analysis based on the size of various parameters in Table 1 as well as computation time in Table 2. In Table 1, " − " represents that a particular parameter is not required. We assume that each authority is responsible for only one attribute. As one can see from Table 1, the proposed scheme supports a constant-length ciphertext. In addition, from Table 2, one can see that the pairing operations also remain constant due to the *constant-length ciphertext* approach. In Table 3, we give the feature-based comparative analysis for the proposed scheme against existing schemes.

Table 1. Size of parameters for multi-authority ABE schemes.

Scheme	MPK	MSK	SK	CT	Expressiveness of policy
$\boldsymbol{r_1}$=# CT attributes,$\boldsymbol{r_2}$=# SK attributes, $\|\boldsymbol{Z}\|$= Z element bit-length, $\|\boldsymbol{G}\|$= G element bit-length, $\boldsymbol{n}$=# system attributes, $\boldsymbol{n'}$=#*OR* gates in the policy					
[8]	$O(1)\|G\|$	$O(1)\|Z\|$	$O(1)\|G\|$	$O(r_1)\|G\|$	Any threshold gate
[9]	$O(n)\|G\|$	$O(n)\|Z\|$	$O(r_2)\|G\|$	$O(r_1)\|G\|$	AND gate
[10]	$O(1)\|G\|$	−	$O(r_2)\|G\|$	$O(r_1)\|G\|$	Any threshold gate
[30]	$O(1)\|G\|$	$O(1)\|G\|$	$O(r_2)\|G\|$	$O(n')\|G\|$	Any threshold gate
[31]	−	−	$O(n)\|G\|$	$O(r_1)\|G\|$	Any threshold gate
[16]	$O(n)\|G\|$	$O(n)\|Z\|$	$O(r_2)\|G\|$	$O(1)\|G\|$	AND gate
[17]	$O(1)\|G\|$	$O(1)\|G\|$	$O(r_2)\|G\|$	$O(1)\|G\|$	Any threshold gate
[18]	$O(n)\|G\|$	$O(n)\|Z\|$	$O(r_2)\|G\|$	$O(1)\|G\|$	Any threshold gate

[19]	$O(1)\|G\|$	–	$O(r_2)\|G\|$	$O(1)\|G\|$	AND gate
[11]	$O(1)\|G\|$	–	$O(r_2)\|G\|$	$O(r_1)\|G\|$	Any threshold gate
[12]	$O(1)\|G\|$	$O(1)\|G\|$	$O(r_2)\|G\|$	$O(n')\|G\|$	Any threshold gate
[13]	–	–	$O(n)\|G\|$	$O(r_1)\|G\|$	Any threshold gate
[14]	–	–	$O(n)\|G\|$	$O(r_1)\|G\|$	Any threshold gate
[15]	–	–	$O(n)\|G\|$	$O(r_1)\|G\|$	Any threshold gate
Our Work	$O(n)\|G\|$	$O(1)\|Z\|$	$O(r_2)\|G\|$	$O(1)\|G\|$	AND gate

Table 2. Computational comparison for proposed schemes.

Scheme	Encryption	Decryption
$\boldsymbol{T_{Exp}}$= One exponent time, $\boldsymbol{T_{mul}}$= One multiplication time, $\boldsymbol{T_{pairing}}$= One pairing time		
[8]	$O(r_1)T_{Exp} + O(1)T_{Mul}$	$O(r_1)(T_{Exp} + T_{Mul} + T_{Pairing})$
[9]	$O(r_1)(T_{Exp} + T_{Mul} + T_{Pairing})$	$O(r_1)(T_{Exp} + T_{Mul} + T_{Pairing})$
[10]	$O(1)T_{Exp} + O(n)(T_{Mul} + T_{Pairing})$	$O(n)(T_{Mul} + T_{Pairing})$
[30]	$O(n'r_1)T_{Exp} + O(r_1)T_{Mul}$	$O(r_1)T_{Mul} + O(1)T_{Pairing}$
[31]	$O(1)(T_{Exp} + T_{Mul} + T_{Pairing})$	$O(r_1)(T_{Mul} + T_{Pairing})$
[16]	$O(n)(T_{Mul} + T_{Pairing})$	$O(1)(T_{Exp} + T_{Pairing})$
[17]	$O(n)(T_{Mul} + T_{Pairing})$	$O(1)(T_{Exp} + T_{Pairing})$
[18]	$O(n)(T_{Mul} + T_{Pairing})$	$O(1)(T_{Exp} + T_{Pairing})$
[19]	$O(r_1)T_{Exp} + O(1)T_{Mul}$	$O(r_1)(T_{Exp} + T_{Pairing})$
[11]	$O(1)T_{Exp} + O(n)(T_{Mul} + T_{Pairing})$	$O(n)(T_{Mul} + T_{Pairing})$
[12]	$O(n'r_1)T_{Exp} + O(r_1)T_{Mul}$	$O(r_1)T_{Mul} + O(1)T_{Pairing}$
[13]	$O(1)(T_{Exp} + T_{Mul} + T_{Pairing})$	$O(r_1)(T_{Mul} + T_{Pairing})$
[14]	$O(n)(T_{Mul} + T_{Pairing})$	$O(n)(T_{Exp} + T_{Pairing})$
[15]	$O(n)(T_{Mul} + T_{Pairing})$	$O(n)(T_{Exp} + T_{Pairing})$
Our Work	$O(1)T_{Exp} + O(r_1)T_{Mul}$	$O(r_1)T_{Mul} + O(1)T_{Pairing}$

Table 3. Feature-based comparative analysis.

Scheme	Multi-authority	Constant-length Ciphertext	Proxy Re-encryption	Scheme	Multi-authority	Constant-length Ciphertext	Proxy Re-encryption
[8]	✓	✗	✗	[18]	✓	✓	✗
[9]	✓	✗	✗	[19]	✗	✓	✗
[10]	✓	✗	✗	[11]	✗	✗	✓
[30]	✓	✗	✗	[12]	✗	✗	✓
[31]	✓	✗	✗	[13]	✗	✗	✓
[16]	✗	✓	✗	[14]	✗	✗	✓
[17]	✗	✓	✗	[15]	✗	✗	✓
Our scheme					✓	✓	✓

6. Conclusion and Future Work

CP-ABE is the efficient technique for the multicasting feature in the security. However, the basic CP-ABE scheme suffers from various important features like ciphertext length. In research, authors have proposed different schemes for each of the features, but none of the schemes provided all of these features. Thus, in this paper, we have proposed a scheme to provide all-in-one features, which makes the proposed scheme applicable in many scenarios as compared to its predecessors. In the future, one can extend the scheme using proxy-based mechanism to make it suitable for cloud-based environments. One can also extend the proposed scheme for the constant-length secret key to reduce the complexity.

References

[1] R. L. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public-key

Cryptosystems," Commun. ACM, vol. 21, no. 2, pp. 120–126, 1978.

[2] A. Shamir, "Identity-based Cryptosystems and Signature Schemes," Proc. of Workshop on the Theory and Application of Cryptographic Techniques (CRYPTO 1984), vol. 196, pp. 47–53, 1984.

[3] A. Sahai and B. Waters, "Fuzzy Identity-based Encryption," Proc. of the Annual Int. Conf. on the Theory and Applications of Cryptographic Techniques (UROCRYPT 2005), vol. 3494, pp. 457–473, 2005.

[4] V. Goyal, O. Pandey, A. Sahai and B. Waters, "Attribute-based Encryption for Fine-grained Access Control of Encrypted Data," Proc. of the 13th ACM Conf. on Computer and Communications Security, pp. 89–98, DOI: 10.1145/1180405.1180418, 2006.

[5] J. Bethencourt, A. Sahai and B. Waters, "Ciphertext-policy Attribute-based Encryption," Proc. of the IEEE Symposium on Security and Privacy (SP'07), pp. 321–334, Berkeley, CA, USA, 2007.

[6] M. Chase, "Multi-authority Attribute Based Encryption," Proc. of Theory of Cryptography Conference (TCC 2007), Part of the Lecture Notes in Computer Science Book Series, vol. 4392, pp. 515–534, 2007.

[7] S. Muller, S. Katzenbeisser and C. Eckert, "On Multi-authority Ciphertext-policy Attribute-based Encryption," Bull. Korean Math. Soc., vol. 46, no. 4, pp. 803–819, 2009.

[8] N. Gorasia, R. R. Srikanth, N. Doshi and J. Rupareliya, "Improving Security in Multi Authority Attribute Based Encryption with Fast Decryption," Procedia Computer Science, vol. 79, DOI: 10.1016/j.procs.2016.03.080, 2016.

[9] V. Božović, D. Socek, R. Steinwandt and V. I. Villányi, "Multi-authority Attribute-based Encryption with Honest-but-curious Central Authority," Int. J. Comput. Math., vol. 89, no. 3, pp. 268–283, 2012.

[10] H. Lin, Z. Cao, X. Liang and J. Shao, "Secure Threshold Multi Authority Attribute Based Encryption without a Central Authority," Proc. of the Int. Conf. on Cryptology in India, pp. 426–436, 2008.

[11] X. Zhang and Y. Yin, "Research on Digital Copyright Management System Based on Blockchain Technology," Proc. of the IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), pp. 2093–2097, Chengdu, China, 2019.

[12] Z. Xu, J. Shen, P. Luo and F. Liang, "PVcon: Localizing Hidden Concurrency Errors with Prediction and Verification," IEEE Access, vol. 8, pp. 165373–165386, 2020.

[13] J. Shen, X. Deng and Z. Xu, "Multi-security-level Cloud Storage System Based on Improved Proxy Re-encryption," EURASIP J. on Wireless Communication and Networking, vol. 2019, no. 1, p. 277, 2019.

[14] Z. Xu, J. Shen, F. Liang and Y. Chen, "Fine-grained Access Control Scheme Based on Improved Proxy Re-encryption in Cloud," J. Adv. Comput. Intell. Intell. Informatics, vol. 25, no. 2, pp. 170–176, 2021.

[15] G. Pareek and B. R. Purushothama, "KAPRE: Key-aggregate Proxy Re-encryption for Secure and Flexible Data Sharing in Cloud Storage," J. of Information Security and Applications, vol. 63, p. 103009, 2021.

[16] R. Kothari, N. Choudhary and K. Jain, "CP-ABE Scheme with Decryption Keys of Constant Size Using ECC with Expressive Threshold Access Structure," Proc. of Emerging Trends in Data Driven Computing and Communications, Part of the Studies in Autonomic, Data-driven and Industrial Computing Book Series, Springer, pp. 15–36, 2021.

[17] Z. Zhang, W. Zhang and Z. Qin, "Fully Constant-size CP-ABE with Privacy-preserving Outsourced Decryption for Lightweight Devices in Cloud-assisted IoT," Security and Commun. Networks, vol. 2021, Article ID 6676862, DOI: 10.1155/2021/6676862, 2021.

[18] Z. Zhang and S. Zhou, "A Decentralized Strongly Secure Attribute-based Encryption and Authentication Scheme for Distributed Internet of Mobile Things," Computer Networks, vol. 201, p. 108553, 2021.

[19] W. Yang, R. Wang, Z. Guan, L. Wu, X. Du and M. Guizani, "A Lightweight Attribute Based Encryption Scheme with Constant Size Ciphertext for Internet of Things," Proc. of the IEEE Int. Conf. on Communications (ICC 2020), 2020, pp. 1–6, Dublin, Ireland, 2020.

20] Y. Zhang, J. Li and H. Yan, "Constant Size Ciphertext Distributed CP-ABE Scheme with Privacy Protection and Fully Hiding Access Structure," IEEE Access, vol. 7, pp. 47982–47990, 2019.

[21] S. F. Tan and A. Samsudin, "Recent Technologies, Security Countermeasure and Ongoing Challenges of Industrial Internet of Things (IIoT): A Survey," Sensors, vol. 21, no. 19, DOI: 10.3390/s21196647, 2021.

[22] C. Ge, Z. Liu, J. Xia and L. Fang, "Revocable Identity-based Broadcast Proxy Re-encryption for Data Sharing in Clouds," IEEE Trans. on Dependable and Secure Comp., vol. 18, no. 3, pp. 1214–1226, 2019.

[23] L. Fang et al., "A Secure and Authenticated Mobile Payment Protocol against off-site Attack Strategy," IEEE Trans. on Dependable and Secure Computing, In Press, DOI: 10.1109/TDSC.2021.3102099, 2021.

[24] C. Ge, W. Susilo, J. Baek et al., "Revocable Attribute-based Encryption with Data Integrity in Clouds," IEEE Trans. on Dependable and Secure Computing, DOI: 10.1109/TDSC.2021.3065999, 2021.

[25] C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia and L. Fang, "A Verifiable and Fair Attribute-based Proxy Re-encryption Scheme for Data Sharing in Clouds," IEEE Trans. on Dependable and Secure Computing, In Press, DOI: 10.1109/TDSC.2021.3076580, 2021.

[26] C. Ge, W. Susilo, Z. Liu, J. Xia, P. Szalachowski and L. Fang, "Secure Keyword Search and Data Sharing Mechanism for Cloud Computing," IEEE Trans. on Dependable and Secure Computing, vol. 18, no. 6, pp. 2787–2800, 2020.

[27] F. Guo, Y. Mu, W. Susilo, D. S. Wong and V. Varadharajan, "CP-ABE with Constant-size Keys for Lightweight Devices," IEEE Trans. on Inf. Forensics and Security, vol. 9, no. 5, pp. 763–771, 2014.

[28] Y. Chen, L. Song and G. Yang, "Attribute-based Access Control for Multi-authority Systems with Constant Size Ciphertext in Cloud Computing," China Communications, vol. 13, no. 2, pp. 146-162, 2016.

[29] W. Susilo, G. Yang, F. Guo and Q. Huang, "Constant-size Ciphertexts in Threshold Attribute-based Eencryption without Dummy Attributes," Information Sciences, vol. 429, pp. 349–360, 2018.

[30] A. Lewko and B. Waters, "Decentralizing Attribute-based Encryption," Proc. of the Annual Int. Conf. on the Theory and Applications of Cryptographic Techniques (EUROCRYPT), vol. 6632, pp. 568–588, 2011.

[31] S. Müller, S. Katzenbeisser and C. Eckert, "Distributed Attribute-based Encryption," Proc. of the Int. Conf. on Information Security and Cryptology (ICISC 2008), vol. 5461, pp. 20–36, 2008.

[32] M. Blaze, G. Bleumer and M. Strauss, "Divertible Protocols and Atomic Proxy Cryptography," Proc. of the Int. Conf. on the Theory and Applications of Cryptographic Techniques, vol. 1403, pp. 127–144, 1998.

[33] M. Mambo and E. Okamoto, "Proxy Cryptosystems: Delegation of the Power to Decrypt Ciphertexts," IEICE Trans. Fundam. Electron. Commun. Comput. Sci., vol. 80, no. 1, pp. 54–63, 1997.

[34] G. Ateniese, K. Fu, M. Green and S. Hohenberger, "Improved Proxy Re-encryption Schemes with Applications to Secure Distributed Storage," ACM Trans. Inf. Syst. Secur., vol. 9, no. 1, pp. 1–30, 2006.

[35] M. Green and G. Ateniese, "Identity-based Proxy Re-encryption," Proc. of the Int. Conf. on Applied Cryptography and Network Security (ACNS 2007), vol. 4521, pp. 288–306, 2007.

[36] T. Matsuo, "Proxy Re-encryption Systems for Identity-based Encryption," Proc. of the International Conference on Pairing-based Cryptography (Pairing 2007), vol. 4575, pp. 247–267, 2007.

[37] S. Guo, Y. Zeng, J. Wei and Q. Xu, "Attribute-based Re-encryption Scheme in the Standard Model," Wuhan Univ. J. Nat. Sci., vol. 13, no. 5, pp. 621–625, 2008.

[38] X. Liang, Z. Cao, H. Lin and J. Shao, "Attribute Based Proxy Re-encryption with Delegating Capabilities," Proc. of the 4th Int. Symp. on Information, Computer and Communications Security, pp. 276–286, 2009.

[39] L. Ibraimi, M. Asim and M. Petković, "An Encryption Scheme for a Secure Policy Updating," Proc. of the Int. Conf. on E-Business and Telecommunications, pp. 304–318, DOI: 10.5220/0002994703990408, 2010.

[40] S. Luo, J. Hu and Z. Chen, "Ciphertext Policy Attribute-based Proxy Re-encryption," Proc. of the Int. Conf. on Information and Communications Security (ICICS 2010), vol. 6476, pp. 401–415, 2010.

[41] K. Emura et al., "A Ciphertext-policy Attribute-based Encryption Scheme with Constant Ciphertext Length," Proc. of the Int. Conf. on Inform. Security Practice and Experience, vol. 5451, pp. 13–23, 2009.

[42] J. Herranz, F. Laguillaumie and C. Ràfols, "Constant Size Ciphertexts in Threshold Attribute-based Encryption," Proc. of the Int. Workshop on Public Key Cryptography (PKC), vol. 6056, pp. 19–34, 2010.

[43] C. Delerablée and D. Pointcheval, "Dynamic Threshold Public-key Encryption," Proc. of the Annual Int. Cryptology Conf., vol. 5157, pp. 317–334, [Online], Available: https://hal.inria.fr/inria-00419154, 2008.

ملخص البحث:

يعدّ التّشفير تقنيةً يتمّ بموجبها تحويل نصٍّ عاديٍ الى نصٍّ مشفّر حتى لا يتمكن المهاجم من استعادة النصّ دون المفتاح السِّرّي. ويُعدّ التّشفير المرتكز على السياسة القائمة على السِّمات إحدى تقنيات التّشفير الموجهة نحو السِّمة التي تتمثل في أنّ المستخدم لا يمكنه فكّ تشفير الرّسالة إلّا إذا تمت تلْبية متطلبات سياسة السِّمات المذكورة في النصّ المشفّر من قِبل سِمات المفتاح السري للمستخدِم. في الدراسات السابقة، إرتجل المؤلفون التّقنية القائمة من أجل تحسين المخطط البسيط لسياسة التّشفير القائم على السِّمات. وحديثاً، في عام 2021، اقترح وانغ وآخرون سياسة التّشفير القائمة على السِّمات باستخدام إعادة التّشفير بالوكالة، وأشاروا الى فعاليتها مقارنةً بالتّقنيات السابقة. إلّا أنّها تتبع النّصّ المشفّر ذا الطّول المتغيّر، الذي يزداد فيه حجم النصّ المشفّر بزيادة عدد السِّمات. كذلك فإنّ تلك التّقنية تقود الى تكاليف تتعلّق بالحوسبة عند المستقبِل خلال فكّ التّشفير الذي يتمّ القيام به عبر أجهزة إنترنت الأشياء.

لذا، اقترحت في هذه الورقة طريقة محسَّنة للحصول على نصّ مشفّرثابت الطّول باستخدام إعادة التّشفير بالوكالة؛ من أجل تقليل وقْت الحوسبة والاتّصال. والجدير بالذّكر أنّ الطّريقة المقترحة تتّسم بالأمان تحت مسألة ديفي هلمان المتعلّقة بالقرارات ثنائية الخطّية.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 08, No. 03, September 2022.

DATA HIDING TECHNIQUE FOR COLOR IMAGES USING PIXEL VALUE DIFFERENCING AND CHAOTIC MAP

Nisreen I. R. Yassin

(Received: 13-Feb.-2022, Revised: 12-Apr.-2022, Accepted: 27-Apr.-2022)

ABSTRACT

The huge advance in information technology and communication has resulted in the growing usage of digital networks, which consequently handed an important role to information security. Steganography is the art of hiding secret message bits into different multimedia data to provide the transferred information with security against unauthorized access. Most techniques applying the pixel value differencing (PVD) approach depend on the sequential embedding manner that lacks security. This study proposes a method that uses a complex chaotic map to randomly choose the coefficients for embedding a secret message. First, the cover image is transformed through integer wavelet transform (IWT). The embedding process starts in the highest-frequency band of IWT and continues to the next subbands according to the message size. Adaptive embedding is then performed depending on the intensity variation between pixel pairs using PVD and least significant bit substitution. The nonsequential embedding performed using the chaotic map makes the method more secure. The experimental results show that the proposed technique achieves a high peak signal-to-noise ratio with an improved capacity compared with other techniques.

KEYWORDS

Data hiding, Integer wavelet transform, Pixel value differencing, Chaotic map.

1. INTRODUCTION

Nowadays, a massive amount of information is being exchanged through the internet. Malicious users constantly try to steal information while performing transfers through that opened network. Information theft can cause systems to fail and countries to become defeated. Therefore, information security has become one of the most important topics to attract the attention of researchers. Cryptography [1]-[2], steganography [3], and watermarking [4] are the most popular technologies used to secure information. In cryptography, the sender changes the shape of the secret data, then using certain tools, the recipient returns the data back to its original shape. Although cryptography guarantees data confidentiality, the encrypted data can be traced and defined during transmission. This has resulted in an increase on the demand for data hiding techniques. Meanwhile, steganography is a technique of hiding secret data into another digital content, such as an image, a video, and an audio. The concealed secret data should not be seen by the naked eye [5]. Steganography is essential in security applications and in private and secure communication. Image steganography techniques are the most common, as image contains many redundant zones and is the most widely used media over the networks [6].

A steganography system consists of a cover image, which is the original image, the secret data hidden in the original image, a stego image that is the original image after hiding the secret data in it, and a security key for better security [3]. Image steganography techniques can be categorized into two types: spatial and frequency domain-based techniques. In the former, the secret data are directly concealed in the cover media. The most popular spatial domain technique is the least significant bit (LSB) substitution. In this approach, the LSBs of the cover media are replaced by the secret data bits [7]-[8]. Another technique is the pixel value differencing (PVD) technique, which hides secret bits in the difference between two consecutive pixels [9]-[10]. Spatial embedding techniques have high embedding capacity and imperceptibility with low computational time and robustness against attacks [11]. In frequency domain-based techniques, the secret data are embedded in the frequency transformation of the cover media. Transformation techniques have higher computational time and higher robustness against attacks but have a lower embedding capacity. There are many transformation techniques used in data hiding such as discrete cosine transform [12] and discrete wavelet transform [4]. The major defiance

N. Yassin is with Department of Systems and Information, Engineering and Renewable Energy Research Institute, National Research Centre, Cairo, Egypt. Email: nisreen.yassin20@gmail.com

in steganography is the preservation of the statistical properties and the visual quality of the cover image after the secret data have been embedded.

The quality of an image steganography system depends on three essential features, that is, image imperceptibility, payload capacity, and robustness. A tradeoff exists between these features; therefore, selecting the suitable embedding locations inside the cover image and identifying the amount of data to be embedded are important for achieving balance. Most steganography techniques select the cover image pixels sequentially to embed the secret data (e.g., PVD techniques). This manner of selection increases their vulnerability to statistical attacks. Chaotic map-based steganography techniques allow the embedment of the secret data in the cover image pixels in a nonsequential mode; however, they usually embed 1 bit of secret data into the cover image pixel, which limits the embedding capacity. Therefore, combining PDV techniques with chaotic map techniques could improve both security against statistical attacks and limited capacity. Embedding the secret data in sharp-edge regions is also less noticeable using human visual system compared with embedding them in smooth and uniform regions. Correspondingly, integer wavelet transform (IWT) is used to embed the secret data in the edge regions of the cover image.

This study proposes a new image steganography technique that applies PVD using a chaotic map for pixel selection. Unlike usual PVD methods, the proposed method embeds the secret data into nonsequential selected image pixels. Random pixel selection is performed using a complex chaotic map, which is very sensitive to the initial conditions and control parameters. The contributions of the proposed technique are as follows:

1- PVD is performed on the basis of a complex chaotic map, in which the selection of common pixels is completely random. Nonsequential embedding increases robustness to statistical attacks.

2- Adaptive embedding based on the contrast difference between the coefficient pairs is proposed.

3- High embedding capacity and efficiency are achieved with results that are comparable with those of other studies in the literature.

The remainder of this paper is organized as follows: Section 2 presents the related materials and methods; Section 3 introduces the proposed system; Section 4 discusses the experimental results; and Section 5 provides the conclusions.

2. RELATED MATERIALS AND METHODS

2.1 Related Materials

Many image steganography techniques in the literature consider spatial and frequency domains. The method of selecting pixels for embedding the secret data is important in attaining efficiency. Adaptive selection, edge detection selection, random selection, and color-dependent selection aim to increase the embedding capacity while preserving the good quality of the stego image. Both the PVD and LSB methods are used to embed the secret data in gray and colored images. The PVD method is a spatial domain method proposed by Wu et al. [9]. In this method, the gray cover image is divided into nonoverlapping blocks. The difference between the two successive pixels in each block is calculated and used to embed the secret data. The PVD method has been enhanced by many researchers to improve its hiding capacity [13]-[15]. In Paul et al. [16], PVD with nonsequential embedding was recently introduced. Pixel pairs were formed by taking the horizontal and vertical pixels from alternatively selected blocks. Although the embedding capacity was increased, the block selection depended on the predefined tables. Mandal et al. [17] proposed a data hiding technique based on the interpolation and difference expansion method. Although the proposed technique achieved a high embedding capacity, the gained peak signal-to-noise ratio (PSNR) was low.

Sahu et al. [18] proposed two reversible data hiding techniques for grayscale images to improve the embedding capacity without sacrificing the image quality. The first technique used LSB matching in dual images, whereas the second technique embedded the secret data using n-rightmost bit replacement. Solak and Altınışık [19] proposed a two-step data hiding scheme based on LSB. The secret information was first encrypted using keywords and shifting. Next, two types of adaptive LSB+3 were proposed to hide the encrypted data into the cover image. Solak [20] proposed a hybrid data hiding technique based on LSB substitution and enhanced modified signed digit algorithms. In the proposed technique, the n-adjacent pixels gained from the cover image and the k-least significant bits are used to embed the secret

data. Paul et al. [21] proposed a method for concealing the secret data in highly energetic pixels to increase robustness against statistical attacks. Liao et al. [22] investigated uniform and adaptive embedding. Setiadi [23] improved payload capacity in LSB using Canny and Sobel edge detection techniques. Feng et al. [24] proposed a method for minimizing the embedding distortion using the syndrome-trellis code. Chakraborty et al. [25] identified the edge areas using an edge predictor to conceal more secret data in sharper edges.

Liao et al. [26] proposed a method for adaptively partitioning the payload capacity among the red–green–blue channels of the cover image based on exploring the interchannel correlations. The method searches for high embedding probabilities and then modifies the pixel costs of the three channels. Liao et al. [27] also proposed a method for adaptively distributing the payload capacity among multiple images based on texture features for multiple-image steganography. They presented two strategies. The first strategy was based on the image texture complexity, whereas the second one was based on the distortion distribution. Al-Qwider et al. [28] proposed a hybrid security system based on gathering cryptography and steganography techniques. The secret message and the stego key were first encrypted using the modified Jamal encryption algorithm. The encrypted message was then hidden in the least 3–3–2 bits of the red–green–blue components of the cover image. Laffont et al. [29] proposed an RGB image steganography technique based on the modulus function. Each pixel value was modified to conceal one digit of the secret data. Wang et al. [30] proposed a watermarking technique for color images based on discrete cosine transform and just noticeable distortion. Abraham et al. [31] proposed a color image watermarking scheme using spatial domain methods. Parah et al. [32] proposed a spatial domain watermarking technique based on inter block pixel difference.

Saraireh et al. [33] proposed an image steganography system based on Haar-DWT. First, the secret message was encrypted using the advanced encryption standard algorithm. Next, the encrypted message was inserted in the DWT subbands. Valandar et al. [34] proposed an image steganography method based on IWT and chaotic map. First, IWT was applied on the cover image. Subsequently, the coefficients were randomly selected using a modified logistic map. Valandar et al. [35] also proposed a steganography technique based on a three-dimensional (3D) sine chaotic map. The secret message was embedded in the LL subband of IWT of an RGB cover image using the random numbers generated from the map. This algorithm embedded only one secret bit in each pixel of the cover image, resulting in a low embedding capacity. Ghebleh et al. [36] proposed a steganography system that concealed the secret data in the lifted discrete wavelet transform of the cover image. The secret data were randomly scattered in the cover image using a 3D chaotic map. Last, Sharafi et al. [37] proposed a new hybrid chaotic map used to present an image steganography method. Wavelet transforms were applied on the cover and secret images using different types of shift operators to enhance resistance against different attacks.

2.2 Methods

2.2.1 IWT

Dealing with digital images requires IWT due to the nature of image pixels, which are integer samples. Wavelet transform outputs are floating point numbers, even if the inputs are integers. Wavelet coefficients are rounded to integers, resulting in errors during the reconstruction of the original image from its transform version. A perfect reconstruction is achieved by eliminating the rounding error using a lifting scheme [38]-[39]. If $\boldsymbol{a}$ and $\boldsymbol{b}$ are two consecutive pixels, IWT can be computed by first computing the difference between $\boldsymbol{a}$ and $\boldsymbol{b}$ and then using that difference to compute the average as described in Equation (1). The inverse lifting is described in Equation (2) [40].

$$d = a - b \quad , \; s = \frac{d}{2} + b \tag{1}$$

$$b = s - \frac{d}{2} \quad , \; a = d + b \tag{2}$$

2.2.2 Chaotic Map

Most hiding techniques use a chaotic behavior to securely embed the secret information in the cover data. Chaotic maps are sensitive to the primary state where small variations in the input may produce large variations in the output. In the proposed method, the complex chaotic map introduced by Ayubi et al. [41] is used to select the positions for the secret data embedment. It is represented in Equations (3)–(5) as follows:

$$[z_1(n+1) \equiv \left(\alpha * \left(z_1(n)/z_2(n)\right)^2 + c_1\right)]\text{CFOLD1} \quad (3)$$

$$[z_2(n+1) \equiv (\beta * (z_2(n)/z_1(n))^2 + c_2)]\ \text{CFOLD 1} \quad (4)$$

$$(z\ CFOLD\ 1) = z^{real}\ Mod1 + (z^{imj} Mod\ 1) \times 1i \quad (5)$$

where z_1, z_2, c_1 and c_2 are complex numbers and $\alpha\ and\ \beta$ are integer numbers between $[10, \infty]$. This chaotic map is used for color images, where the real and imaginary parts of z_1 represent the image coordinates. The real part of z_2 is the color channel. For more security, the imaginary part of z_2 can be used to encrypt the secret data.

3. PROPOSED SYSTEM

Steganography aims to suppress secret messages into digital media without triggering trepidation. Embedding secret data in sequential pixels makes the steganography technique weak against statistical attacks [42]. In this section, we present a steganography technique that utilizes PVD and LSB to embed the secret data into an RGB cover image. The usual PVD divides the cover image into sequential pixel blocks for embedding the secret data. Our proposed technique randomly selects pixels nonsequentially using a chaotic map. The details of the proposed technique are presented in the subsequent sections.

3.1 Preprocessing

The following preprocessing steps are performed before the embedding process:

Step 1. The steganography system inputs are a cover image C of size $M \times N \times 3$, random data with different sizes used as a secret message SM to be hidden in the cover image, and a secret key K.

Step 2. Initialize the chaotic map control parameters $(z_1, z_2, c_1, c_2, \alpha, and\ \beta)$.

Step 3. Extract the RGB color channels of C. IWT is applied on each $R, G, and\ B$ color channel of size $M \times N$. Four subbands $[LL, LH, HL, HH]$ of size $M/2 \times N/2$ are obtained for each color channel.

Step 4. Save the original sign of the IWT coefficients of the cover image as an array of $[1, -1]$ to be used in image restoration.

Step 5. Collect the same subbands of the transformed coefficients of the three color channels together in a cell array $\{HH_{R,G,B}; HL_{R,G,B};\ LH_{R,G,B}; LL_{R,G,B}\}$.

3.2 Embedding Process

Steganography techniques concentrate on the maximization of the embedding capacity without contributing any visual image degradation. Traditional PVD methods sequentially embed a secret message in adjacent pixels by dividing the cover image into blocks. This embedding manner increases the detection ability by tracing the pixels of each block. In the proposed technique, nonsequential embedding is performed through two steps. The first step ensures that all edges are specified by applying IWT. All IWT subbands are used to conceal the secret message; however, the embedding process starts in high-frequency subbands $[HH, HL, LH]$ that represent the image's edge information. In the second step, a chaotic map is used to randomly select pixels from the IWT subbands. A complex chaotic map, which is very sensitive to the initial conditions and control parameters, is used to increase security and randomness. Figure 1 depicts the embedding process. The detailed steps are presented as follows:

Step 1. Start with the cell $HH_{R,G,B}$. Initialize a flag array of 0s converted to 1s after embedding to avoid a repeated selection of coefficients.

Step 2. Two consecutive coefficients $p_{x.y.z}$ and $p_{x+1.y.z}$ are selected from the transformed coefficients of the selected cell using the position generated by the real and imaginary parts of z_1 of the complex chaotic map. The color channel of each selected coefficient is indicated by the real part of z_2.

Step 3. The difference d between the two coefficients $p_{x.y.z}$ and $p_{x+1.y.z}$ is computed. If the value of d is <15, which indicates low variations between the two consecutive coefficients, then the secret bits are embedded through LSB substitution. Three bits are embedded into each selected coefficient from the LSB direction by applying the XOR function between the secret message bits and the generated secret key to obtain the new $p_{x,y,z}^{`}$ and $p_{x+1,y,z}^{`}$ coefficients.

Step 4. If d is >15, PVD is used to conceal the secret bits. The number of secret bits embedded in the difference value d is specified by classifying the value of d into four groups (Table 1). The number of secret bits n and the corresponding lower bound l are obtained according to the d value. Start the embedding process using PVD by converting the secret bits n into the corresponding decimal value c.

1- A new difference $\boldsymbol{d'}$ is calculated as indicated in Equation (6). The absolute difference between the old and new differences is calculated to conceal the secret message in the difference values of the wavelet coefficients. A new value d'' is obtained using Equation (7) and used to acquire possible new wavelet coefficients as declared in Equations (8 and 9).

$$d' = l + floor(c/2) \tag{6}$$

$$d'' = floor\big((abs(d - d')/2)\big) \tag{7}$$

$$p\grave{}_{x,y,z} = p_{x,y,z} \pm d'' \tag{8}$$

$$p\grave{}_{x+1,y,z} = p_{x+1,y,z} \pm d'' \tag{9}$$

2- Two values of new coefficients are obtained for each original coefficient. The new coefficients $p\grave{}_{x,y,z}$ and $p\grave{}_{x+1,y,z}$ are selected as the values nearest to the original coefficients as indicated in Equation (10).

$$p\grave{}_{x,y,z} = \begin{cases} p_{x,y,z} + d'' & if\ \left|p_{x,y,z} - (p_{x,y,z} + d'')\right| < \left|p_{x,y,z} - (p_{x,y,z} - d'')\right| \\ p_{x,y,z} - d'' & otherwise \end{cases}$$

$$p\grave{}_{x+1,y,z} = \begin{cases} p_{x+1,y,z} + d'' & if\ \left|p_{x+1,y,z} - (p_{x+1,y,z} + d'')\right| < \left|p_{x+1,y,z} - (p_{x+1,y,z} - d'')\right| \\ p_{x+1,y,z} - d'' & otherwise \end{cases} \tag{10}$$

The newly obtained coefficients $p\grave{}_{x,y,z}$ and $p\grave{}_{x+1,y,z}$ are referred to as the stego coefficients. Their values must be between 0 and 255.

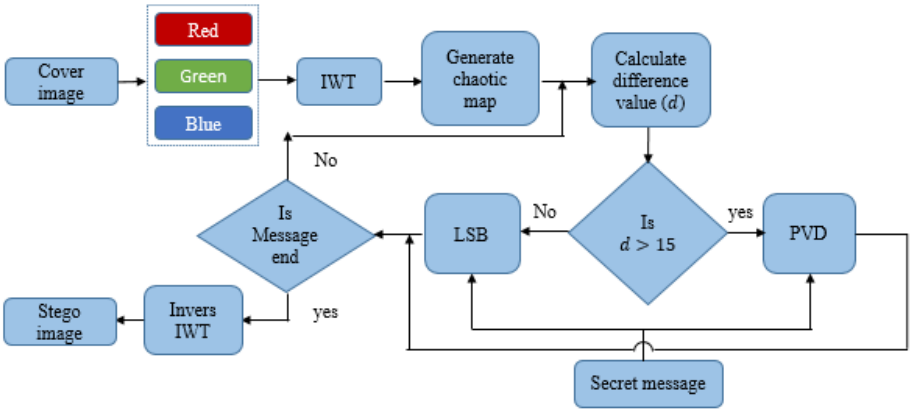


Figure 1. Embedding process of the proposed method.

Step 5. Steps 2–4 are repeated until the secret message bits are embedded. If the secret message has not ended, and all possible pairs in $HH_{R,G,B}$ have been selected, the embedding process restarts from step 1 using the next cell $HL_{R,G,B}$ and so on for cells $LH_{R,G,B}$ and $LL_{R,G,B}$. $LL_{R,G,B}$ is a low-frequency subband; hence, the number of bits in Table 1 is reduced by only 1 bit in the case of PVD embedding to minimize the distortion in this band.

Table 1. Difference groups and corresponding number of secret bits.

Difference Groups [l-u]	G_1 [16-31]	G_2 [32-63]	G_3 [64-127]	G_4 [128-255]
No. of secret bits	4	5	6	7

3.3 Postprocessing

The embedding process is completed by ending the message or by selecting all coefficient pairs in all subbands.

Pseudo-code 1: Embedding process

Input: Cover image C, Secret message SM, Secret key K, Initial values $z_1, z_2, c_1, c_2, \alpha, \beta$.
Output: Stego image I′

(R, G, B) ⟵ C % RGB color channels
[LL, LH, HL, HH] ⟵ IWT [(R, G, B)] % IWT subbands
[a b] ⟵ size(LL) % height and width of subbands
for each R, G, B
  if [LL, LH, HL, HH] < 0 then
    sign ⟵ -1 else sign ⟵ 1
  end
end
bandcell ⟵ $\{HH_{R,G,B}; HL_{R,G,B}; LH_{R,G,B}; LL_{R,G,B}\}$
[h w d] ⟵ size($HH_{R,G,B}$)
for counter =1 to 4
  bandnew ⟵ bandcell(counter)
  flag ⟵ zeros(h, w, d)
  while $s < SM$ do
    $z_1 \longleftarrow (\alpha * (z_1/z_2)^2 + c_1)$
    $z_1 \longleftarrow \mathrm{mod}\left(\left(real(z_1)\right), 1\right) + \mathrm{mod}\left(\left(imag(z_1)\right), 1\right)$*1i
    $z_2 \longleftarrow (\beta * (z_2/z_1)^2 + c_2)$
    $z_2 \longleftarrow \mathrm{mod}\left(\left(real(z_2)\right), 1\right) + \mathrm{mod}\left(\left(imag(z_2)\right), 1\right)$*1i
    x ⟵ $\mathrm{mod}(round(real(z_1) \times 10^{14}), h)$
    y ⟵ $\mathrm{mod}(round(imag(z_1) \times 10^{14}), w)$
    z ⟵ $\mathrm{mod}(round(real(z_2) \times 10^{14}), d) + 1$
    if flag(x, y, z) == 0 && flag (X+1, y, z) == 0 then
      p_1 ⟵ abs(bandnew(x, y, z))
      p_2 ⟵ abs(bandnew(x + 1, y, z))
      $diff$ ⟵ abs($p_1 - p_2$)
        if $diff > th$ then
          [p_{1new} p_{2new} s] ⟵ pvdembed($SM, diff, s, p_1, p_2$) % Apply Eqs.(6:10)
        else
          [p_{1new} p_{2new} s] ⟵ lsbembed(SM, s, p_1, p_2) % 3-LSb replacement
        end
      bandnew(x, y, z) ⟵ p_{1new}
      bandnew(x + 1, y, z) ⟵ p_{2new}
    end
    flag(x, y, z) ⟵ 1, flag(x+1, y, z) ⟵ 1
  end
  bandcell(counter) ⟵ bandnew
end
Reconstruct image
Apply inverse IWT
Get stego image I′

Step 1. The image is reconstructed after the embedding process. The original sign array is then used to give every stego coefficient its original sign by multiplying element-by-element the absolute of the stego array $I^{`}$ and the original sign array I as follows:

$$I^{`}(x,y,z) = abs\left(I^{`}(x,y,z)\right) * I(x,y,z) \tag{11}$$

Step 2. Finally, apply inverse IWT to obtain the stego image $I^{`}$. Pseudo-code 1 illustrates the embedding process in details.

3.4 Extraction Process

The same chaotic map initial conditions and secret key used in the embedding process are required to successfully extract the secret message. The extraction process starts with reading the stego image and transforming it into IWT. The chaotic map is then initialized, and a flag matrix is formed to prevent coefficient reselection. Figure 2 depicts the extraction process. The extraction steps are summarized as follows:

Step 1. The stego image is frequency-transformed using IWT. The chaotic map is initialized. Two consecutive coefficients ($p_{x.y.z}$ and $p_{x+1.y.z}$) are selected using the initialized chaotic map.

Step 2. The difference between the two coefficients d is computed. Three bits are directly extracted from the rightmost direction of the coefficients if d is <15.

Step 3. The lower bound l is identified according to the range of the difference value if d is >15 (Table 1). The decimal value of the secret bits is calculated as follows:

$$c = 2(d - l) \tag{12}$$

Finally, the secret binary bits are obtained by transforming the decimal value c into the binary form and concatenating all of them to acquire the secret message back. Pseudo-code 2 illustrates the extraction process.

Pseudo-code 2: Extraction process

Input: Stego image I′, Secret message size, Secret key K, Initial values $z_1, z_2, c_1, c_2, \alpha, \beta$.
Output: Secret message

(R, G, B) ⟵ I′ % RGB color channels
[LL, LH, HL, HH] ⟵ IWT [(R, G, B)] % IWT subbands
bandcell ⟵ $\{HH_{R,G,B};\ HL_{R,G,B};\ LH_{R,G,B};\ LL_{R,G,B}\}$
[h w d] ⟵ size $(HH_{R,G,B})$
for counter =1 to 4
  flag ⟵ zeros (h, w, d)
  while $s < SM$ do
    z_1 ⟵ $(\alpha * (z_1/z_2)^2 + c_1)$
    z_1 ⟵ $\mathrm{mod}\left((real(z_1)), 1\right) + \mathrm{mod}\left((imag(z_1)), 1\right)$*1i
    z_2 ⟵ $(\beta * (z_2/z_1)^2 + c_2)$
    z_2 ⟵ $\mathrm{mod}\left((real(z_2)), 1\right) + \mathrm{mod}\left((imag(z_2)), 1\right)$*1i
    x ⟵ $\mathrm{mod}(round(real(z_1) \times 10^{14}), h)$
    y ⟵ $\mathrm{mod}(round(imag(z_1) \times 10^{14}), w)$
    z ⟵ $\mathrm{mod}(round(real(z_2) \times 10^{14}), d) + 1$
    if flag(x, y, z)== 0 && flag(x + 1, y, z) == 0 then
      p'_1 ⟵ abs(bandcell(x, y, z))
      p'_2 ⟵ abs(bandcell(x + 1, y, z))
      $diff$ ⟵ p'_1 - p'_2
        if $diff > th$ then
          extract using Table 1 and Eq. (12)
        else
          extract 3-LSB from each p'_1 and p'_2
        end
    end
    flag(x, y, z) ⟵ 1, flag(x+1, y, z) ⟵ 1
  end
end

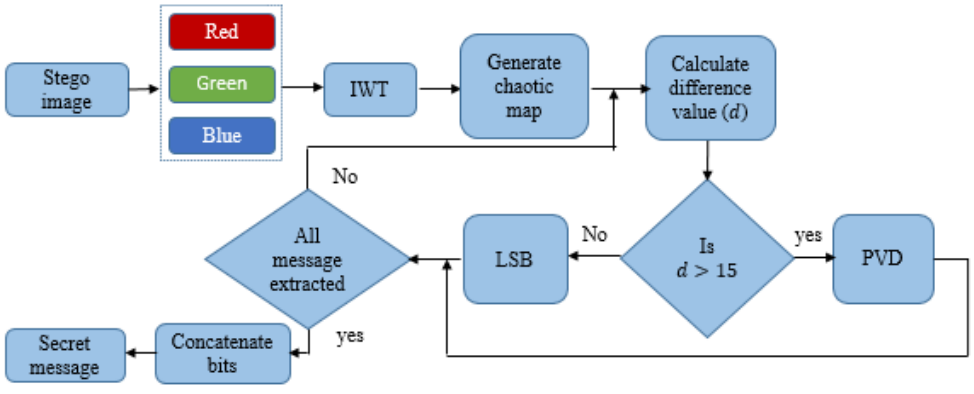


Figure 2. Extraction process of the proposed method.

4. EXPERIMENTAL RESULTS

Many experiments were conducted to evaluate the performance of the proposed technique. Matlab®2017 software on an Intel®Core™i5 CPU at 3.10 GHz computer was used for the simulation process. Standard color images (4.2.01, Lena, baboon, pepper, boat, and house) measuring 512 × 512, which were downloaded from the SIPI image database [43] were used to evaluate and compare the results of the proposed technique with those of the other existing steganography techniques. Figure 3 illustrates the cover images. Random data with different sizes were used as the secret messages.

Three standard metrics were used to evaluate the performance of any proposed steganography technique. The first metric is the embedding efficiency that evaluates the stego image quality. The second one is the payload embedding capacity defined as the amount of secret bits that can be hidden into the cover image. The third metric is the technique's robustness against different image processing operations and attacks.

- Embedding Efficiency Evaluation

The embedding efficiency is assessed using the PSNR, mean square error (MSE), and structural similarity index measure (SSIM) [4, 44]. The PSNR measures the difference between the cover image and its stego version and is computed using Equation (13). The MSE is the error between cover and stego images obtained using Equation (14). The system quality increases as long as the PSNR increases and the MSE decreases.

$$PSNR = 10\log_{10}\left(\frac{(255)^2}{MSE}\right) \tag{13}$$

$$MSE = \frac{1}{M\times N\times 3}\sum_{i=1}^{M}\sum_{j=1}^{N}[I`(i,j) - I(i,j)]^2 \tag{14}$$

where, I is the cover image of size $M \times N \times 3$ and $I`$ is the stego image.

Although the MSE and PSNR measure the absolute error between two images, the SSIM [35] measures the structural similarity between two images and is calculated by Equation (15).

$$SSIM(I, I`) = (2u_x u_y + c_1)(2\sigma_{xy} + c_2)/(u_x^2 + u_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2) \tag{15}$$

where, x and y are two windows of common size; u_x and u_y are the averages of x and y, respectively; σ_{xy} is the co-variance of x and y; σ_x^2 is the variance of x; σ_y^2 is the variance of y; and c_1 and c_2 are two variables used to stabilize the division with a weak denominator.

SSIM result is a value between 0 and 1, where 0 indicates no structural similarity, whereas 1 indicates full structural similarity. Table 2 presents the MSE, PSNR, and SSIM values for some test images at different values of embedding capacities. The proposed technique achieved a high PSNR (i.e., above 50 dB for 125-kB embedding capacity). The achieved PSNR was greater than 30 dB, which is the PSNR threshold [45]. All achieved SSIM values were approximately 1, indicating the quality of the proposed system. Figure 4 shows the stego images after concealing 1,472,164 secret bits. The stego images clearly showed no degradation and were visually identical to the cover images, from which the naked eye cannot observe any difference between the cover and stego images. In summary, the proposed technique exhibited a good embedding efficiency.

Table 2. Results of the PSNR, MSE, and SSIM.

Cover image	**39kB**			**116kB**			**125kB**			**192kB**		
	PSNR	MSE	SSIM	PSNR	MSE	SSIM	PSNR	MSE	SSIM	PSNR	MSE	SSIM
4.2.01	56.46	0.14	0.99	51.76	0.43	0.99	51.43	0.46	0.99	49.54	0.72	0.99
Lena	57.12	0.12	0.99	52.24	0.38	0.99	51.94	0.41	0.99	50.12	0.63	0.99
baboon	52.82	0.33	0.99	48.15	0.99	0.99	47.86	1.06	0.99	46.01	1.62	0.99
pepper	56.85	0.13	0.99	52.21	0.39	0.99	51.89	0.420	0.99	50.00	0.64	0.99
boat	56.09	0.15	0.99	51.26	0.48	0.99	50.95	0.52	0.99	49.15	0.79	0.99
house	56.53	0.14	0.99	51.77	0.43	0.99	51.45	0.46	0.99	49.58	0.71	0.99

Figure 3. Standard cover images: 4.2.01, Lena, baboon, pepper, boat, and house (left to right).

Figure 4. Stego images: 4.2.01, Lena, baboon, pepper, boat, and house (left to right).

- Embedding Capacity Evaluation

The embedding capacity is the maximum amount of secret data that can be hidden in the cover without being noticed. It is also known as the embedding rate that can be calculated using Equation (16).

$$Embedding\ rate = \frac{Number\ of\ embedded\ secret\ bits}{Image\ size} \quad (16)$$

Figure 5 illustrates the efficiency of the proposed system at different embedding rates, where the PSNR is plotted at the embedding rates of 20%, 40%, 60%, 80%, and 100% for three colored images (i.e., Lena, pepper, and house). For the 20% embedding rate, the PSNR was greater than 45 dB and decreased as the embedding rate increased until approximately 35 dB at 100% embedding rate (1,578,972 bits). The proposed system achieved considerably high embedding rates with an acceptable PSNR. Additionally, the used images exhibited approximately the same results according to different embedding rates.

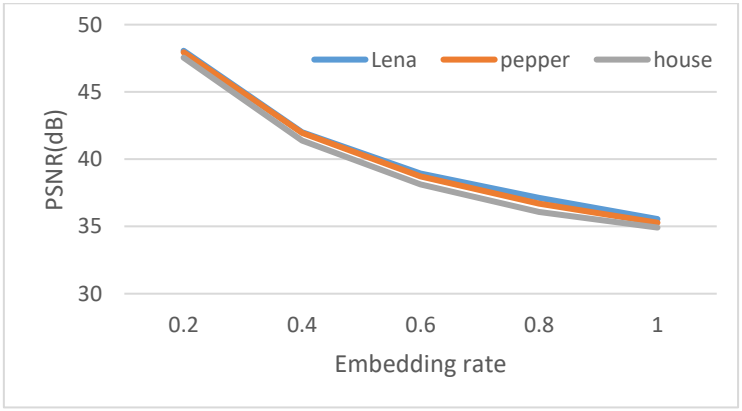


Figure 5. Efficiency of the proposed system at different embedding rates.

- Robustness Evaluation

Figure 6 shows a histogram plot of the RGB color channels of the Lena cover image and the Lena stego image to determine the robustness of the proposed system against a histogram analysis. The histogram form for the three colored channels was conserved with very minimal changes. Figure 7 plots and fits the histogram of the difference between the cover and stego images as a normal distribution. The histogram of the difference values is concentrated around zero, indicating the robustness of the proposed system.

We used the Virtual Steganographic Laboratory (VSL) tool to test the proposed technique against steganalysis. The VSL tool is a steganography detection software used to apply the RS analysis, which estimates the concealed message's length. Figure 8 presents a sample report of the VSL tool showing the name of the input stego image and the corresponding detected message size in bytes. For the used images, the ratio between the average detected capacities to the actual capacity was computed. The obtained detection ratio was 20%, indicating the robustness of the proposed technique.

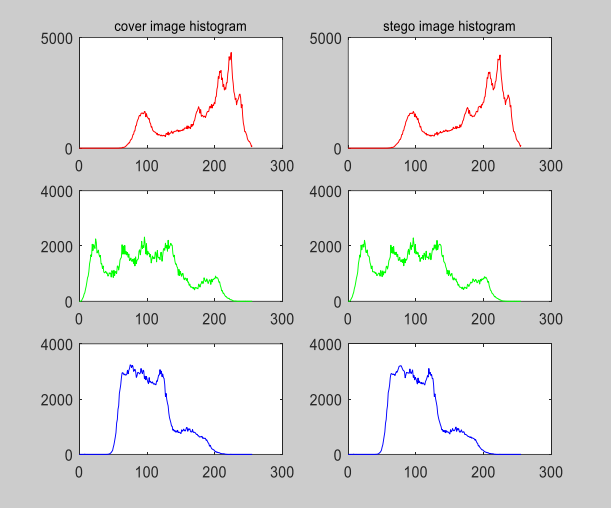


Figure 6. Histogram of the Lena cover and stego images.

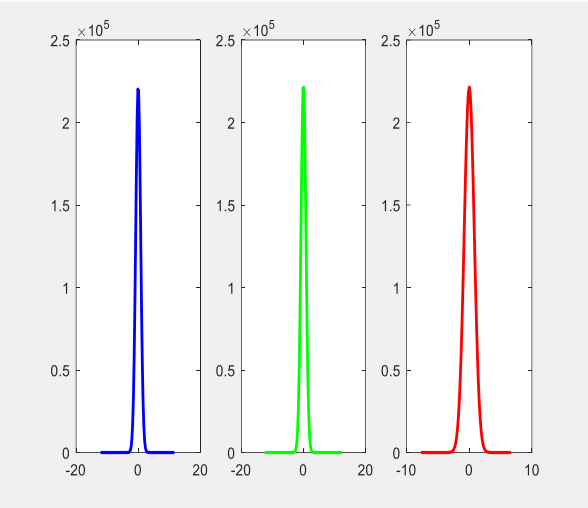


Figure 7. Histogram of the difference between the Lena cover and stego images.

	A	B	C	D	E
1	1	LSB-RS	E:\testfolder_2\stego_Lena.png	512x512	Estimated message size [B]:773.1707724709962
2	2	LSB-RS	E:\testfolder_2\stego_Baboon.png	512x512	Estimated message size [B]:5891.683894314265
3	3	LSB-RS	E:\testfolder_2\stego_boat.png	512x512	Estimated message size [B]:1446.253526212815
4	4	LSB-RS	E:\testfolder_2\stego_house.png	512x512	Estimated message size [B]:1025.732603759307
5	5	LSB-RS	E:\testfolder_2\stego_Pepper.png	512x512	Estimated message size [B]:9317.495468119538
6	6	LSB-RS	E:\testfolder_2\stego_4.2.01.png	512x512	Estimated message size [B]:1777.9765822308163
7	7	LSB-RS	E:\testfolder_2\stego_jet.png	512x512	Estimated message size [B]:557.4034805599041

report

Figure 8. VSL sample report.

- Computational Complexity

Computational complexity is important for evaluating the performance of any algorithm. It depends on the processor speed, available memory, and algorithm type. Table 3 presents the processing time of the proposed technique. The obtained results were compared with those proposed in [34] using the same cover image and embedding capacities.

Table 3. Processing time of the proposed technique.

Cover image	Secret message (kb)	Embedding time (s)		Extraction time (s)	
		Proposed technique	[34]	Proposed technique	[34]
Lena	Random data (39)	0.286129	0.827536	0.138638	0.469742
Lena	Random data (116)	0.628100	1.568793	0.487249	1.391331
Lena	Random data (125)	0.656901	1.596450	0.419632	1.403308

- Comparison of the Proposed Technique with Other Techniques

Some algorithms based on IWT that used the same color images and implemented chaotic maps were used to compare our technique with that in the literature. Table 4 shows the comparison results of the proposed technique and those proposed in [34] and [36] according to the embedding capacity, PSNR, and SSIM. Although the PSNR values reported in [34] and [36] were higher than those achieved by the proposed system at the same payload capacity, we found a high amelioration in the embedding capacity; that is, 1,578,972 bits can be hidden without any degradation in the image with approximately 38 dB.

Table 5 presents the comparison results of the proposed technique and those presented in [35] and [29]. In [35], 77,244 bits were embedded as the secret data in the cover images (Lena, baboon, pepper, and jet). Compared with the technique presented in [35], the proposed technique achieved higher PSNR and SSIM values for Lena, pepper, and jet but lower values for baboon. In [29], different sizes of secret data were embedded into the cover images. Table 5 shows that the results accomplished using the proposed technique were greater than or approximately equal to those achieved in [29].

Table 4. Comparison of the proposed technique with methods in the literature.

Cover image	**Proposed technique**			**[34]**			**[36]**		
	capacity	PSNR	SSIM	capacity	PSNR	SSIM	capacity	PSNR	SSIM
Lena	125 kb	51.944	0.9998	125 kb	52.998	0.9979	125 kb	56.135	0.9997
baboon	125 kb	47.868	0.9994	125 kb	52.993	0.9993	NA	NA	NA

Table 5. Comparison of the proposed technique with methods in the literature.

Cover image	Secret size (bits)	**Proposed technique**		**[35]**		Secret size (bytes)	**Proposed technique**		**[29]**	
		PSNR	SSIM	PSNR	SSIM		PSNR	SSIM	PSNR	SSIM
Lena	77244	54.1489	0.9999	51.3761	0.9984	83654	41.5132	0.9980	41.2968	NA
baboon	77244	49.9544	0.9997	52.4228	0.9991	91286	41.2815	0.9949	40.8535	NA
pepper	77244	54.0779	0.9999	52.3907	0.9986	78612	42.0032	0.9981	41.5641	NA
jet	77244	53.7585	0.9998	51.9846	0.9980	81851	41.1846	0.8893	41.2968	NA

With relevance to the PVD-based methods, the proposed technique was compared with that in [46] as a PVD-based method that uses the same color images with the same size. Table 6 shows that the proposed technique achieved a higher PSNR compared with that in [46] under the same payload capacities for all images. Table 7 presents a comparison of the proposed technique and the algorithms in [22], [24], [25], [21] and [9] according to the payload embedding capacity and the PSNR. The proposed technique had a higher embedding capacity with an acceptable PSNR than the other algorithms. The maximum payload that can be embedded without any image degradation was 2bpp with 37.5 dB PSNR, which proved the superiority of the proposed technique.

Although many techniques apply PVD and achieve a higher capacity than the proposed technique, most of them implement the commonly used sequential embedding approach. On the contrary, the proposed technique selects the pixel pairs in a fully nonsequential manner using a chaotic map that leads to increased security. Embedding the secret data in the IWT subbands also ensures complete reversibility and prefect reconstruction of data.

Table 6. Comparison of the proposed technique with PVD-based methods.

Image 512x512x3	**Payload capacity**	**Embedding rate (bpp)**	**PSNR**	
			PVD [46]	**proposed**
Lena	810757	1.030	37.13	40.31
baboon	918877	1.168	34.95	36.87
pepper	812986	1.033	36.74	39.81
jet	818887	1.041	36.36	39.24
boat	851837	1.083	36.03	38.66
house	834866	1.061	36.57	39.01
Average	841368	1.069	36.30	38.98

Table 7. Average embedding rate (AER) and the PSNR comparison.

Method	**AER (bpp)**	**PSNR (dB)**
Liao et al. [22]	≤ 1	44.72
Feng et al. [24]	≤ 1	26.83
Soumendu et al. [25]	≤ 1	30.0
Paul et al. [21]	≤ 1	38.54
Wu et al. [9]	1.56	39.06
Proposed technique	2.00	37.54

Table 8. Comparing PSNR values using binary secret logos.

Binary logo size	**9×(85×85)**		**64×64**			**3×(64×64)**	
	Proposed	[37]	Proposed	[30]	[31]	Proposed	[32]
Lena	54.9103	54.4578	66.8594	45.4018	53.35	62.4419	40.58
baboon	51.6294	52.1523	63.0485	43.8262	53.35	58.2916	39.60

Table 8 shows a comparison of the proposed technique with recent techniques using binary secret logos. The results indicate that the proposed technique had the superiority according to PSNR and embedding capacity.

5. CONCLUSIONS

In this study, we proposed a new image steganography technique based on the nonsequential application of the PVD approach using a complex chaotic map to withstand statistical attacks. We performed adaptive embedding using LSB or PVD to minimize the embedding distortion. The embedding process started in the highest-frequency detail band of IWT and continued to the other bands until the secret data embedding was completed.

Many experimental tests were conducted to evaluate the performance of the proposed technique. In summary, we obtained the following conclusions:

- According to the embedding efficiency test, the stego image looked identical to the cover image, from which the naked human eye cannot observe any difference between the two images even at high embedding rates.

- The robustness of the proposed technique was achieved. We observed no difference between the histograms of the cover images and the corresponding stego images. Moreover, only 20% of the embedded secret data could be detected using the VSL tool.

- In comparison with other techniques, the proposed technique exhibited a good performance and a good balance between the embedding capacity and imperceptibility.

CONFLICT OF INTERESTS

The author declares that no conflict of interest.

ACKNOWLEDGEMENTS

The author appreciates National Research Centre (NRC), Cairo, Egypt for funding this study through research project No (12010501).

REFERENCES

[1] H. M. Ghadirli, A. Nodehi and R. Enayatifar, "An Overview of Encryption Algorithms in Color Images," Signal Processing, vol. 164, pp. 163-185, 2019.

[2] P. Ayubi, S. Setayeshi and A. M. Rahmani, "Deterministic Chaos Game: A New Fractal Based Pseudo-random Number Generator and Its Cryptographic Application," Journal of Information Security and Applications, vol. 52, p. 102472, 2020.

[3] I. J. Kadhim, P. Premaratne, P. J. Vial and B. Halloran, "Comprehensive Survey of Image Steganography: Techniques, Evaluations and Trends in Future Research," Neurocomputing, vol. 335, pp. 299-326, 2019.

[4] E. M. El Houby and N. I. Yassin, "Wavelet-hadamard Based Blind Image Watermarking Using Genetic Algorithm and Decision Tree," Multimedia Tools and Applications, vol. 79, pp. 28453-28474, 2020.

[5] N. F. Johnson and S. Jajodia, "Exploring Steganography: Seeing the Unseen," Computer, vol. 31, pp. 26-34, 1998.

[6] S. Nazari, A. M. Eftekhari-Moghadam and M. S. Moin, "A Novel Image Steganography Scheme Based on Morphological Associative Memory and Permutation Schema," Security and Communication Networks, vol. 8, pp. 110-121, 2015.

[7] Z. Xia, X. Wang, X. Sun and B. Wang, "Steganalysis of Least Significant Bit Matching Using Multi-order Differences," Security and Communication Networks, vol. 7, pp. 1283-1291, 2014.

[8] M. Afrakhteh and J. A. Lee, "Adaptive Least Significant Bit Matching Revisited with the Help of Error Images," Security and Communication Networks, vol. 8, pp. 510-515, 2015.

[9] D.-C. Wu and W.-H. Tsai, "A Steganographic Method for Images by Pixel-value Differencing," Pattern Recognition Letters, vol. 24, pp. 1613-1626, 2003.

[10] H.-C. Wu, N.-I. Wu, C.-S. Tsai and M.-S. Hwang, "Image Steganographic Scheme Based on Pixel-value Differencing and LSB Replacement Methods," IEE Proceedings-Vision, Image and Signal Processing, vol. 152, pp. 611-615, 2005.

[11] H. Al-Dmour and A. Al-Ani, "A Steganography Embedding Method Based on Edge Identification and XOR Coding," Expert Systems with Applications, vol. 46, pp. 293-306, 2016.

[12] A. Saxena and F. C. Fernandes, "DCT/DST-based Transform Coding for Intra Prediction in Image/Video Coding," IEEE Transactions on Image Processing, vol. 22, pp. 3974-3981, 2013.

[13] A. K. Gulve and M. S. Joshi, "An Image Steganography Algorithm with Five pixel Pair Differencing and Gray Code Conversion," International Journal of Image, Graphics and Signal Processing, vol. 6, p. 12, DOI:10.5815/ijigsp.2014.03.02, 2014.

[14] A. K. Gulve and M. S. Joshi, "An Image Steganography Method Hiding Secret Data into Coefficients of Integer Wavelet Transform Using Pixel Value Differencing Approach," Mathematical Problems in Engineering, vol. 2015, DOI: 10.1155/2015/684824, 2015.

[15] G. Swain, "Two New Steganography Techniques Based on Quotient Value Differencing with Addition-subtraction Logic and PVD with Modulus Function," Optik, vol. 180, pp. 807-823, 2019.

[16] G. Paul, S. K. Saha and D. Burman, "A PVD Based High Capacity Steganography Algorithm with Embedding in Non-sequential Position," Multimedia Tools and Applications, vol. 79, pp. 13449-13479, 2020.

[17] P. C. Mandal, I. Mukherjee and B. N. Chatterji, "High Capacity Reversible and Secured Data Hiding in Images Using Interpolation and Difference Expansion Technique," Multimedia Tools and Applications, vol. 80, pp. 3623-3644, 2021.

[18] A. K. Sahu and G. Swain, "High Fidelity Based Reversible Data Hiding Using Modified LSB Matching and Pixel Difference," Journal of King Saud University-Computer and Information Sciences, vol. 34, no. 4, DOI: 10.1016/j.jksuci.2019.07.004, 2019.

[19] S. Solak and U. Altınışık, "Image Steganography Based on LSB Substitution and Encryption Method: Adaptive LSB+ 3," Journal of Electronic Imaging, vol. 28, p. 043025, DOI: 10.1117/1.JEI.28.4.043025, 2019.

[20] S. Solak, "High Embedding Capacity Data Hiding Technique Based on EMSD and LSB Substitution Algorithms," IEEE Access, vol. 8, pp. 166513-166524, 2020.

[21] G. Paul, I. Davidson, I. Mukherjee and S. Ravi, "Keyless Steganography in Spatial Domain Using Energetic Pixels," Proc. of the International Conference on Information Systems Security (ICISS 2012), vol. 7671, pp. 134-148, 2012.

[22] X. Liao, Z. Qin and L. Ding, "Data Embedding in Digital Images Using Critical Functions," Signal Processing: Image Communication, vol. 58, pp. 146-156, 2017.

[23] D. Setiadi, "Improved Payload Capacity in LSB Image Steganography Uses Dilated Hybrid Edge Detection," J. King Saud Univ.-Comput. Inf. Sci., vol. 34, no. 2, DOI:10.1016/j.jksuci.2019.12.007, 2019.

[24] B. Feng, W. Lu and W. Sun, "Secure Binary Image Steganography Based on Minimizing the Distortion on the Texture," IEEE Transactions on Information Forensics and Security, vol. 10, pp. 243-255, 2014.

[25] S. Chakraborty, A. S. Jalal and C. Bhatnagar, "LSB Based Non Blind Predictive Edge Adaptive Image Steganography," Multimedia Tools and Applications, vol. 76, pp. 7973-7987, 2017.

[26] X. Liao, Y. Yu, B. Li, Z. Li and Z. Qin, "A New Payload Partition Strategy in Color Image Steganography," IEEE Transactions on Circuits and Systems for Video Technology, vol. 30, pp. 685-696, 2019.

[27] X. Liao, J. Yin, M. Chen and Z. Qin, "Adaptive Payload Distribution in Multiple Images Steganography Based on Image Texture Features," IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 2, pp. 897-911, 2020.

[28] W. H. Al-Qwider and J. N. B. Salameh, "Novel Technique for Securing Data Communication Systems by Using Cryptography and Steganography," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 3, no. 2, pp. 110-130, 2017.

[29] A. Laffont, P. Maniriho, A. Ramsi, G. Guerteau and T. Ahmad, "Enhanced Pixel Value Modification Based on Modulus Function for RGB Image Steganography," Proc. of the 11th Int. Conf. on Information & Communication Technology and System (ICTS), 2017, pp. 61-66, Surabaya, Indonesia, 2017.

[30] J. Wang, W. B. Wan, X. X. Li, J. De Sun and H. X. Zhang, "Color Image Watermarking Based on Orientation Diversity and Color Complexity," Expert Systems with Applications, vol. 140, p. 112868, DOI: 10.1016/j.eswa.2019.112868, 2020.

[31] J. Abraham and V. Paul, "An Imperceptible Spatial Domain Color Image Watermarking Scheme," Journal of King Saud University-Computer and Information Sciences, vol. 31, no. 1, pp. 125-133, 2019.

[32] S. A. Parah, J. A. Sheikh, N. A. Loan and G. Bhat, "A Robust and Computationally Efficient Digital Watermarking Technique Using Inter Block Pixel Differencing," in Multimedia Forensics and Security, ed. Springer, pp. 223-252, 2017.

[33] S. M. Saraireh and A. M. Matarneh, "Higher Level Security Approach for Data Communication System Based on AES Cryptography and DWT Steganography," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 2, no. 3, pp. 179-193, 2016.

[34] M. Y. Valandar, P. Ayubi and M. J. Barani, "A New Transform Domain Steganography Based on Modified Logistic Chaotic Map for Color Images," Journal of Information Security and Applications, vol. 34, pp. 142-151, 2017.

[35] M. Y. Valandar, M. J. Barani, P. Ayubi and M. Aghazadeh, "An Integer Wavelet Transform Image Steganography Method Based on 3D Sine Chaotic Map," Multimedia Tools and Applications, vol. 78, pp. 9971-9989, 2019.
[36] M. Ghebleh and A. Kanso, "A Robust Chaotic Algorithm for Digital Image Steganography," Communications in Nonlinear Science and Numerical Simulation, vol. 19, pp. 1898-1907, 2014.
[37] J. Sharafi, Y. Khedmati and M. Shabani, "Image Steganography Based on a New Hybrid Chaos Map and Discrete Transforms," Optik, vol. 226, no. 2, p. 165492, 2021.
[38] N. I. R. Yassin and E. M. F. El Houby, "Image Steganography Technique Based on Integer Wavelet Transform Using Most Significant Bit Categories," International Journal of Intelligent Engineering and Systems, vol. 15, pp. 499-508, 2022.
[39] C. Liji, K. Indiradevi and K. A. Babu, "Integer-to-integer Wavelet TransformBased ECG Steganography for Securing Patient Confidential Information," Procedia Technology, vol. 24, pp. 1039-1047, 2016.
[40] A. La Cour–Harbo and A. Jensen, "Wavelets and the Lifting Scheme," Proc. of Encyclopedia of Complexity and Systems Science, pp. 10007-10031, DOI: 10.1007/978-0-387-30440-3_588, Springer New York, 2009.
[41] P. Ayubi, M. Jafari Barani, M. Yousefi Valandar, B. Yosefnezhad Irani and R. Sedagheh Maskan Sadigh, "A New Chaotic Complex Map for Robust Video Watermarking," Artificial Intelligence Review, vol. 54, pp. 1237-1280, 2021.
[42] N. Provos, "Defending against Statistical Steganalysis," Proc. of the 10th USENIX Security Symposium (USENIX Security 01), [Online], Available: https://www.usenix.org/conference/10th-usenix-security-symposium/defending-against-statistical-steganalysis, 2001.
[43] USC-SIPI Image Database, [Online], Available: http://sipi.usc.edu/database/database.php?volume=misc.
[44] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," IEEE Transactions on Image Processing, vol. 13, pp. 600-612, 2004.
[45] N.-I. Wu and M.-S. Hwang, "Data Hiding: Current Status and Key Issues," IJ Network Security, vol. 4, pp. 1-9, 2007.
[46] S. Shen, L. Huang and Q. Tian, "A Novel Data Hiding for Color Images Based on Pixel Value Difference and Modulus Function," Multimedia Tools and Applications, vol. 74, pp. 707-728, 2015.

ملخص البحث:

أدّى التّقدّم الهائل في تكنولوجيا المعلومات والاتّصالات الى استخدامٍ مُفرطٍ للشّبكات الرّقمية، الأمر الذّي يجعل أمن المعلومات له دور هامّ. لذا، عمدت بعض التّقنيات الى إخفاء رسائل سرّيّة في بيانات وسائل التّواصُل المختلفة باستخدام إمّا الحقْل الحيّزي أو الحقْل التّرددي لتوفير الأمان للمعلومات المنقولة. وتجدر الإشارة الى أنّ غالبية التّقنيات التي تطبّق طريقة التّفريق بين قِيم النُّقط (PVD) تعتمد على أسلوب التّضمين التّتابُعي، الأمر الذي يضرّ بأمْن المعلومات.

في الطّريقة المقترحة في هذا البحث، تُستخدم خريطة فوضوية معقّدة للاختيار العشوائي لأزواج العوامل من أجْل تضمين الرّسالة السّرّية. يتمّ أوّلاً تحويل صورة الغلاف باستخدام تحويل المويجات التّامّ (IWT). بعد ذلك، تبدأ عملية التّضمين في النّطاق التّردّدي الأعلى من تحويل المويجات التّامّ وتستمرّ الى النّطاقات الفرعية التالية. ويتمّ إجراء التّضمين التّكيّفي وفْق تغيُّر الشِّدة بين أزواج النُّقَط باستخدام التّفريق بين قِيم النُّقَط (PVD) واستبدال الأجزاء الأقلّ أهمّية (LSB).

ويجعل التّضمينُ غيْرُالتّتابعيِ الّذي يتمّ بواسطة الخريطة الفَوْضوية الطّريقةَ المقترحة أكثر أماناً. وقد بيّنت النّتائج التّجريبية أنّ التّقنية المقترحة في هذه الورقة حققت نسبة إشارةٍ الى ضجيج عالية إضافةً الى سِعة محسَّنة لدى مقارنتها بطرقٍ مُستخدَمة أُخرى.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 08, No. 03, September 2022.

A NOVEL TRUE-REAL-TIME SPATIOTEMPORAL DATA STREAM PROCESSING FRAMEWORK

Ature Angbera[1] and Huah Yong Chan[2]

(Received: 9-Mar.-2022, Revised: 2-May-2022, Accepted: 23-May-2022)

ABSTRACT

The ability to interpret spatiotemporal data streams in real time is critical for a range of systems. However, processing vast amounts of spatiotemporal data out of several sources, such as online traffic, social platforms, sensor networks and other sources, is a considerable challenge. The major goal of this study is to create a framework for processing and analyzing spatiotemporal data from multiple sources with irregular shapes, so that researchers can focus on data analysis instead of worrying about the data sources' structure. We introduced a novel spatiotemporal data paradigm for true-real-time stream processing, which enables high-speed and low-latency real-time data processing, with these considerations in mind. A comparison of two state-of-the-art real-time process architectures was offered, as well as a full review of the various open-source technologies for real-time data stream processing and their system topologies were also presented. Hence, this study proposed a brand-new framework that integrates Apache Kafka for spatiotemporal data ingestion, Apache Flink for true-real-time processing of spatiotemporal stream data, as well as machine learning for real-time predictions and Apache Cassandra at the storage layer for distributed storage in real time. The proposed framework was compared with others from the literature using the following features: Scalability (Sc), prediction tools (PT), data analytics (DA), multiple event types (MET), data storage (DS), Real-time (Rt) and performance evaluation (PE) stream processing (SP) and our proposed framework provided the ability to handle all of these tasks.

KEYWORDS

Spatiotemporal big data, Real-time processing, Stream processing, Apache Kafka, Apache Flink, Apache Cassandra, Apache Spark.

1. INTRODUCTION

One of the new elements for Internet-based applications is spatial-temporal data. Data utilized for real-time analytics has become a part of production data in new internet application trends [1]. Spatiotemporal data is in enormous quantities through a variety of activities, such as clicks on a social-media platform orders, sales, shipment data in retail and so on. As the internet-connected world grows exponentially, a vast volume of data is generated in a continuous stream from a variety of sources. Five billion people utilize various varieties of mobile devices, according to [2]. Again, according to an IBM big-data study, there will be around 35 zettabytes of data generated yearly from 2020 [3] and data growth will be 50 times faster than it is presently [4]. Every day, 2.5 quintillion bytes of information are created [3]. The amount of data that can be generated is limitless. This phenomenon is referred to as "big data" [1]. In conjunction with this concept, the 3V's of big data (Volume, Velocity and Variety) have been coined [5]. The first V stands for the massive amounts of data produced by these technologies. The second V represents the rate at which the sources generate these data, while the third V represents the data's heterogeneity. There is usually a large amount of spatial-temporal data [6] that is being generated. The 3.8 billion individuals and 8.06 billion internet-connected devices are responsible for the vast amounts of data produced [7]. According to [8], one zettabyte of data was generated in 2010 and seven zettabytes in 2014. As a result of the rapid emergence of massive spatiotemporal datasets, volume, variety and velocity are all concerns that must be addressed. The phrase "spatiotemporal big-data volume" refers to a large number of data that necessitates a great amount of processing and storage [9]. The volume of data is increasing faster than computational processing devices can keep up [10]. Spatiotemporal data is a continuous stream of data in terms of velocity. As a result, the concerned stakeholders have to spend a lot of money on processing [11]. Real-time data stream processing is critical for a variety of applications, but processing a massive volume of data coming from several sources, like online traffic, sensor networks and many other sources, is a significant issue [3]. The most serious problem has been that the spatiotemporal big-data

1. A. Angbera is with School of Computer Sciences, Universiti Sains Malaysia, Pulau Pinang 11800, Malaysia. And with Department of Computer Science Joseph Sarwuan Tarka University, Makurdi, Nigeria. Email: angberaature@student.usm.my
2. H. Y. Chan is with School of Computer Sciences, Universiti Sains Malaysia, Pulau Pinang 11800, Malaysia. Email: hychan@usm.my

system is based around "Hadoop"; namely "MapReduce" [12]-[13]. This framework has a high level of scalability and fault tolerance. A vast volume of data in batches can be managed by this system and it provides observation blow understanding of past data, but it can only analyze a constrained collection of data. Although MapReduce [14]–[19] isn't designed for processing real-time stream applications, its critical function is to process data as soon as it arrives to get a quick response and make effective decisions. As a result, for effective and speedy analysis having very low latency and high throughput, a real-time stream processing spatiotemporal data novel framework is required.

From the literature, it is observed that there is a vast growth of spatiotemporal datasets which are coming from different sources, resulting in different structures of spatiotemporal datasets. This leads to a serious problem known as heterogeneity of spatiotemporal data [6]. Processing of these spatiotemporal datasets is challenging, as they do not have a common data structure presentation for modeling, resulting in inaccurate results during data analytics. Also, it was observed that spatiotemporal big-data systems are based around “Hadoop”, which is not designed to process data in real time; hence, in our study, we proposed a true-real-time stream processing framework for spatiotemporal data, that takes care of the heterogeneous data issues and processes big spatiotemporal data in real time.

This paper gives an overview of certain essential concepts in spatiotemporal unbounded data, stream processing, big-data storage and other related fields. After that, we present some tools and systems that enable real-time data processing. A full comparison of various queueing message systems, stream processing platforms and data storage platforms is also made available. In addition, we offer a novel framework based on the previous comparison. Finally, we compare and contrast our suggested novel framework with others from the literature. The rest of the paper is structured as follows: Section 2 highlights spatiotemporal big data. Data-processing technologies are presented in Section 3. Section 4 presents the distributed queuing management technologies with a comparison among them. Section 5 presents the technologies for big-data storage. The state-of-the-art real-time processing architectures are highlighted and compared in Section 6. Section 7 presents the proposed framework and Section 8 provides the conclusions.

2. SPATIOTEMPORAL BIG DATA

In terms of spatial data storage, geographic analysis and spatiotemporal visualization, a lot of big-data platforms fared poorly [20]. In industry, research and a range of other fields, the term "big data" is now commonly used. "Spatiotemporal big data" describes the creation of huge datasets, which are unable to be analyzed and managed due to the vast amount and complexity of data collected at a certain given time. As information technology has progressed, the demand for processing, understanding and displaying spatiotemporal large data has increased significantly. A vast volume of geographical and spatiotemporal data, as well as its application sectors, necessitate a distributed and highly scalable data-processing system [21]. This demand is being met by researchers from both academia and industry. The MapReduce framework, Hadoop [22], NoSQL databases [23] and Spark [24] are all used in modern large-scale geographic data processing systems. The majority of these systems improved previous systems by adding spatial or spatiotemporal support as a layer on top of them or by extending the core of them. A high number of these systems were built from the ground up or run on platforms other than Hadoop, NoSQL and Spark. A true-real-time stream processing spatiotemporal data framework having these properties; namely, scalability (Sc), prediction tools (PT), data analytics (DA), multiple event types (MET), data storage (DS), Real-time (Rt) and performance evaluation (PE) stream processing (SP), is rare in the literature.

- Stream Processing (SP): To process data that just comes in, the system should use stream processing. Stream-processing technologies enable us to transform and analyze data as it is being received. As a result, stream processing is critical for delivering real-time functionality [25].
- Data Storage (DS): To save essential data, the system needs to include a data-storage layer. This data can be used in other operations as well as for historical analyses. Furthermore, storage systems should be able to store a vast amount of data as a result of the size of spatiotemporal data [26].

- Performance Evaluation (PE): To demonstrate the proposed system's correctness and effectiveness, an assessment should be available. In many cases, performance is critical; as a result, it should've been evaluated to present enterprises with adequate knowledge to measure the success of the suggested solution and its suitability for the current circumstances [25].
- Multiple Event Types (MET): The technology should support a wide range of kinds of events as input sources, including various forms, such as XML, Java Map and JSON and also as unstructured forms, such as text, CSV and actual data. This characteristic will enable the framework to manage and incorporate a diverse set of spatiotemporal data sources, allowing it to be used in real-world settings [25].
- Scalability (Sc): Modern large-data stream processing engines stress scalability as a fundamental quality attribute [27]. The system must be scalable; this is to aid a high volume of data control. In most cases, more data sources can be incorporated into the same infrastructure and this should be able to handle the influx of new spatiotemporal data [25].
- Prediction Technologies (PT): The technology could also provide developers with prediction tools to help them enhance the level of service and the outcomes [25].
- Data Analytics (DA): This spatiotemporal (heterogeneous) data must be analyzed by at least one mechanism in the system. The capacity to evaluate the data we collect to uncover and report circumstances of interest to interested agents is one of the most significant benefits of the internet of things [26].
- Real-time Processing (RP): Spatiotemporal data should be processed in real time or close to real time. This kind of system will be able to respond to situations of interest as early as possible and as effectively as possible if we can conduct all of the supplied functionality immediately [25]-[26].

3. DATA PROCESSING TECHNOLOGIES

3.1 Data Stream Processing

Real-time data analytics with low latency and high throughput needs became increasingly important in many sectors, such as healthcare, transportation and smart homes [28]. In the industry, stream processing is getting a lot of popularity as a new programming paradigm for implementing real-time data-driven applications [29]. “A stream is an infinite series of tuples in a distributed data stream processing system (DSPS). A data source reads data from an external source (or sources) and feeds it into the system as streams of data. A processing unit (PU) takes tuples from data sources or other PUs and processes them with user-supplied code. It can then transfer the data to other PUs for further processing [30]. To express parallelism, a DSPS typically uses two levels of abstraction (logical and physical). An application is typically depicted as a directed graph in the logical layer, with each vertex corresponding to a data source or a PU and direct edges indicating how data tuples are transmitted between data sources/PUs. Each data source or PU can run as many parallel jobs as possible on a cluster of machines and each task is an instance of that data source or PU. A DSPS's physical layer typically consists of a group of virtual or physical machines that process data received and a master that acts as the cluster's central control unit, distributing user code, scheduling jobs and monitoring them for problems. An application graph is run on numerous worker processes on multiple (physical or virtual) machines at runtime. In most cases, each machine is set up with many slots. The number of slots specifies how many worker processes can execute on this machine and can be pre-configured by the cluster operator based on hardware constraints (such as the number of CPU cores)”. Each worker process has its slot, which is used to process data tuples using user code utilizing one or more threads. Normally, at runtime, a job is assigned to a thread (even if it does not have to be this way). A scheduling mechanism in a DSPS outlines how threads are assigned to processes and machines. A default scheduler is included with many DSPS; however, it can be modified with a custom scheduler. The default scheduler often employs a straightforward scheduling approach that distributes threads to pre-configured processes, which are subsequently assigned to machines in a round-robin fashion. This technique results in a nearly even workload distribution throughout the cluster's available machines. In addition, a DSPS usually provides multiple grouping options, which specify how tuples are distributed among tasks [30].

3.2 Stream Processing Platforms

In this part, we explore and present the differences between data stream processing tools, such as Apache Spark, Apache Hadoop, Apache Storm and Apache Flink. To organize and analyze data, classic relational database management systems, as well as many current batch processing tools, like Hadoop and Spark, have been deployed. Although these technologies have progressed and are beneficial for several products, they are not the greatest choice for creating real-time applications [31]. As a result, emerging innovations, like Apache Storm, Apache Flink and others, have been developed to manage vast quantities of data streams, process them and analyze them as they move to accomplish the demands of real-time applications. These technologies strive to capture the importance of time in real-time analytics, streaming analytics and sophisticated-event processing. We are inspired to provide a truly real-time stream processing framework for spatiotemporal data because of the necessity of such emerging technologies. We will show that earlier techniques, such as MapReduce, do not provide real-time processing despite their capacity to process a vast volume of data, not minding the rate at which the data comes in.

Apache Spark: Spark is a unified large-data analytics engine with built-in streaming, SQL, machine learning and graph-processing modules. It was created at the University of California in 2009, released as an open source in 2010 and given to the Apache Software Foundation in 2013, which has been in charge of the project since then. Spark is the successor to Hadoop, which was the original big-data analytics platform and was used for batch processing [32]. The MapReduce paradigm typically employs a linear data flow to take data out of the disc, map a function across the data, reduce the results to that map and ultimately save this reduced result on the disc-inspired Spark. Furthermore, Spark's "Resilient Distributed Dataset" (RDD) enables multiple readings of datasets as well as interactive data analysis [33]. The Spark adds in-memory processing, which allows for up to 100 times quicker processing, albeit it has the drawback of requiring smaller datasets than Hadoop due to resource constraints. Spark's architecture is made up of Spark Core, which is the project's foundation and the modules or frameworks listed above, which are built on top of it: MLlib for machine learning, Spark Streaming, Spark SQL and GraphX for graph processing. Through an API based on the RDD abstraction, Spark Core provides basic I/O functionality as well as distributed task dispatching and scheduling. This construction is depicted in detail in Figure 1.

Apache Hadoop: It's a platform with open access for data processing that makes use of commodity technology to store and analyze enormous volumes of data. The Hadoop ecosystem is seen in Figure 2 along with the framework's major components. The "Hadoop Distributed File System" (HDFS) and the "MapReduce programming" style are the two most significant components of the Hadoop architecture. The data is stored in HDFS and processed in a distributed way using MapReduce. Despite its many benefits, Hadoop lacks storage and network encryption, has limited flexibility, is unsuitable for tiny-data collections and has a large I/O overhead. Hadoop, particularly the Map-Reduce framework, which is never the better technology for processing the most recent set of data, is constrained to batch processing. This is one of its major disadvantages [3].

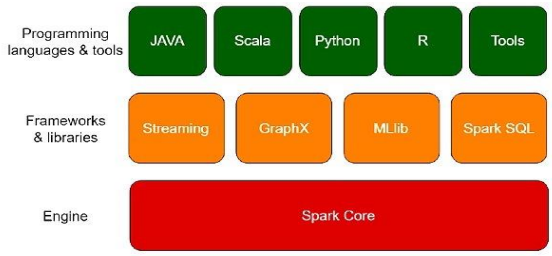


Figure 1. Apache Spark architecture.

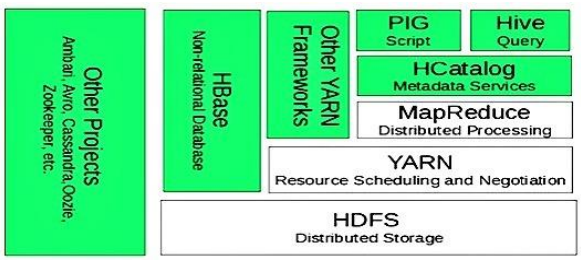


Figure 2. Hadoop ecosystem [3].

Apache Storm: It's an open-source distributed framework that makes it easier to create fault-tolerant programs that run in parallel on computing clusters [34]. The Storm was developed by BackType, a business that was acquired by Twitter in 2011. It is an Eclipse Public License-compliant open-source project. In Storm, a topology is a computing network (Figure 3) that defines how data (such as tuples) travels between processing units [35]. A topology can continue to run indefinitely or until it is interrupted by a user. Similar to earlier application designs, a topology gathers information and

separates it into portions that are handled by assignments to cluster nodes. Data that nodes share is tuples, which are sorted collections of values. The Storm is built on a master-slave paradigm, with a master node running the Nimbus daemon and keeping a membership list to ensure data-processing reliability. According to Nimbus, it connects to Apache Zookeeper [35].

A Storm cluster is comprised of 3 nodes, as illustrated in Figure 4: "Nimbus," (when the original Nimbus instance fails, the secondary Nimbus instance takes over [36]). That is the same as Hadoop's job tracker, "Supervisor," which is in charge of starting and halting the process and "Zookeeper," a common coordination server that governs the cluster of the Storm [3].

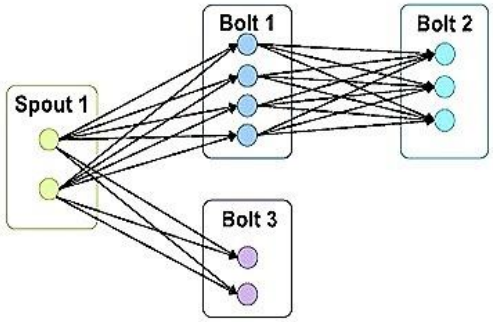


Figure 3. Topology of a storm.

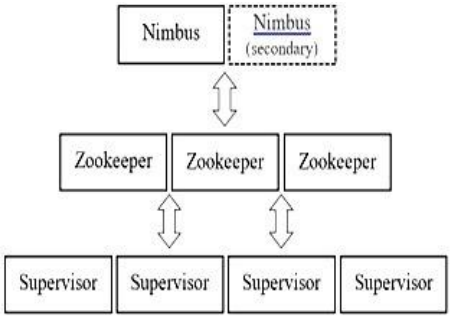


Figure 4. Storm Architecture [36].

Apache Flink: It's a platform for stream and batch data processing with open access, that arose from a fork of the "Stratosphere" project, which was founded in 2010 and developed by a team of researchers from Humboldt-Universität zu Berlin, Technical University Berlin and Hasso-Plattner-Institut Postdam with funding from the German Research Foundation. The project's goal was to develop a new big-data analytics platform to aid research in Berlin-area universities. It was elevated to a high-stage project at the "Apache Software Foundation" at the end of 2014 [32]. The master-slave model is the base design for Flink, which is made up of three primary components. Job Manager: It is the distributed execution's coordination node (master node) that manages the data flow between the slave nodes' task managers. The Task Manager is in charge of executing the operators that receive and produce streams, notifying the Job Manager of their status and exchanging data streams amongst the operators (task managers). Client: It converts computer code into a data-flow graph, which is then sent to the Job Manager to be executed. Flink is a native (true) stream-processing framework that can also handle batch processing, considering each batch as a stream of bounded data. Apache Flink combines stream processing with CEP (Complex-event Processing) [37] technology to provide real-time data analysis and response. Flink allows us to apply transformations to data streams and then analyze the results [38]. Figure 6 shows the ecosystem of flink.

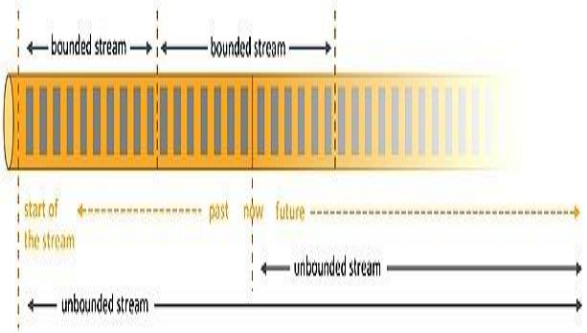


Figure 5. Structure of streams [32].

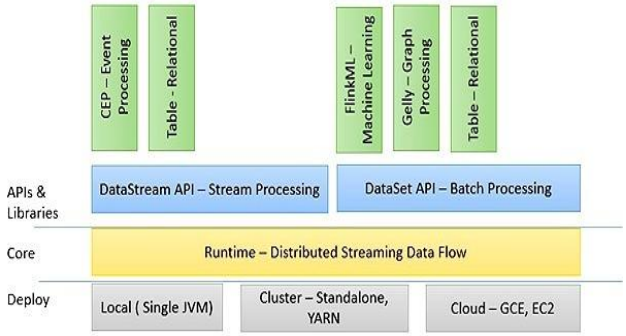


Figure 6. An ecosystem of Flink.

Streams that are unbounded and bounded are depicted in detail in Figure 5. Unbounded streams have a beginning, but no finish; and to achieve a complete result, the sequence in which the events are generated frequently matters. Bounded streams have a beginning and an end, but they may be sorted; thus, the order of events isn't important. Batch processing is the term for this method.

Table 1 lists the characteristics of the data-processing tools mentioned in this paper. For stream processing, Flink is a better technology for true real-time processing, Hadoop handles batch

processing and Spark can manage micro-batching, according to the comparison of several streaming data processing platforms offered in Table 1. To minimize the latency overhead that batching and micro-batching impose, Storm uses the spouts and bolts to execute one-at-a-time processing. Flink supports batch and true stream processing. It's highly optimized, with features such as light-weighted snapshots and it appears to be the data stream management system, the market leader. As we can see, the majority of the desired features (low latency, high throughput, guarantee of exactly-once execution and state management) are available. As a result, we adopt Flink as our computational framework for processing streaming data in our study.

Table 1. Data-processing technologies.

Features	**Apache Spark**	**Apache Flink**	**Apache Hadoop**	**Apache Storm**
Open access	Yes	Yes	Yes	Yes
Coordination tool	Zookeeper	Zookeeper	Zookeeper	Zookeeper
Language	Python, R, C#, Scala, Java	Scala, Python, SQL, Java	Scala, Python, Java	Any PL
In-memory processing	Yes	Yes	No	Yes
Data processing	Batch/Stream (micro-batch)	Batch/Stream (Native)	Batch	Streaming
Execution model	Micro-batch	Real-time (Ture streaming), micro-batch and batch.	Batch	Real-time (one at a time)
Fault tolerance	Yes	Yes	Yes	Yes
Achievable latency	Low latency	Lowest latency as compared with Spark and Storm	High	Very low latency
Data-processing guarantee	Exactly-once processing	Exactly-once processing	Exactly-once processing	At least once processing
Data storage	Yes	Yes	Yes	No
Optimization	Manual	Automatic	Manual	Manual
Operating system	Windows, macOS, Linux	Linux, macOS, Windows	UNIX, Windows	Windows, Linux, macOS
Throughput	High	Very high	Very low	Low

Flink has similar features to Spark, but it operates as a native stream engine, posing numerous obstacles to Spark in stream processing (e.g. in the case of latency and recovery). Flink also appears to be stronger than Storm [39].

4. DISTRIBUTED QUEUING MANAGEMENT TECHNOLOGIES

Data is transferred from one program to another using a messaging system. Applications can concentrate just on data rather than on how it is exchanged. Traditional messaging systems exist, but the majority of them are incapable of working with a huge volume of data in a real-time setting. Message queuing reliability is a key feature of distributed messaging systems. The P to P (point to point) pattern and the publish-subscribe pattern are the two types of message patterns. In a messaging system, the publish-subscribe, commonly known as pub-sub, is used [1]. Publish/subscribe messaging has been supported by distributed queue management solutions, like RabbitMQ, Amazon Kinesis, Kafka and Google Pub/Sub in recent years [31]. When it comes to transferring massive amounts of data around for real-time applications, these technologies have provided some beneficial new solutions. While distributed queue management systems may appear to be identical to traditional message queuing technologies, their architecture is vastly different and as a result, their performance and behavioral properties are vastly different. Traditional queuing schemes, for example, eliminate handled responses out from the queue and are unable to spread out when multiple consumers perform different activities at the same time. Distributed queuing systems, on the other hand, are well-suited for both online and offline content ingestion, because they can accommodate numerous clients and prevent data loss by distributing resilient discs across replicated clusters. The responses are committed to the dispersed queues as soon as feasible, ensuring message delivery for a set amount of time. Each distributed queue management solution splits its topics (i.e., where a producer publishes data

(messages) and a consumer retrieves it). The messages are absorbed by every consumer segment (partitions) of a specific subject, with just a single consumer from the same consumer segment consuming the same partition. The consumer group's function is quite beneficial for re-balancing when partitions and/or customers change [31]. Table 2 lists aspects to consider when selecting a distributed queuing system, involving messaging guarantees, disaster recovery, replication, federated queues (which disperse a single queue's load across nodes or clusters), supported languages and many others.

Table 2. Distributed message queuing technologies.

Features	**Kinesis**	**Ms. Azure Event Hub**	**Apache Kafka**	**RabbitMQ**	**Google pub/sub**
Supported language	Java, Python, .NET, C++, Go, PHP, Ruby, Node.js	Java, C++, Ruby, PHP, Node.js, Python, .NET	PHP, Ruby, Java, Python, .NET, Node.js, Go, C++	Go, C++/C, Java, Python, .NET, PHP, Ruby, Node.js	PHP, Ruby, Java, C++, Node.js, Python, .NET
Messaging guarantees	Yes / At least once	Yes / At least once	Yes / At least once	Yes / At least once	Yes / At least once
Configurable persistence period	from one to seven days (default is 24 hours)	24 hours as default (from one to seven days)	No maximum	N/A	Seven days (non-configurable) or only when it is recognized by all subscribers
Latency	200 ms to 5 seconds	There are no values cited.	Some set-ups are measured in ms. Benchmarking revealed a median delay of ~2 ms.	There are no values cited.	There are no values cited.
Recovery of disaster	Yes	Yes	Yes	Yes	Yes
Replication	Hidden (across three zones)	Configurable replicas	Configurable replicas	Configurable replicas	Hidden
Consumer groups	Yes	Yes	Yes	Yes	Yes
Guarantees ordering	Guaranteed within the confines of a partition	Guaranteed within the confines of a partition	Guaranteed within the confines of a partition	Guaranteed using AMQP channel	No order guarantees
Throughput	1 MB/s input, 2 MB/s output or 1000 records per second can all be supported by a single shard. 20,000 messages per second throughput	Throughput units have been scaled. Each one can handle 1 MB/s entrance, 2 MB/s egresses and 84 GB of storage. The standard tier allows for a total of 20 throughput units.	30,000 messages per second throughput	There are no figures for throughput that have been mentioned.	The standard is 100 MB/s in and 200 MB/s out; however, the maximum speed is stated to be infinite.

Apache Kafka is a real-time communication system that uses a distributed publish-subscribe model. Kafka can handle a large volume of data, allowing you to send messages at the end-point. We also choose Kafka over other popular messaging systems in this study for three reasons: To begin with, other similar message broker technologies, such as RabbitMQ, Amazon Kinesis, ActiveMQ and other enterprise messaging systems, are ephemeral, meaning that they keep data in memory or other light storage. Kafka, on the other hand, provides durability by persisting data on storage, which expands and broadens its application scenarios. Second, Kafka is a data-transit technology rather than a data-processing system. This distinguishes it from the competition in stream processing. The third reason is that Kafka is frequently used in conjunction with other systems for streaming-data processing.

For ingestion, Apache Kafka is currently state-of-the-art. To consume Kafka, two sets of actors are

required, as shown in Figure 7. **Producers** distribute messages on one or more Kafka topics. Data is sent to Kafka brokers by the producers. When a producer sends a message to a broker, it is considered published. Producers have the option of sending messages to a certain partition. **Consumers** are in charge of pulling data from Kafka brokers and sending it to processing nodes (e.g. Spark or Flink). Kafka brokers are managed and coordinated by a Zookeeper. When the latest broker is deployed to the Kafka system or when a broker in the Kafka system fails, the Zookeeper service is used to notify producers and consumers.

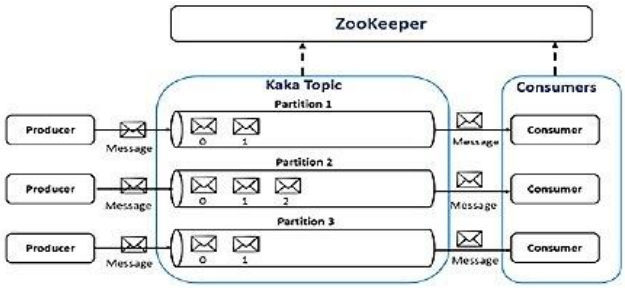


Figure 7. Apache Kafka framework.

5. TECHNOLOGIES FOR BIG DATA STORAGE

The difficulty of huge spatiotemporal data quantities that are growing at an exponential rate has lately been addressed by upgrading big-data analysis tools. The big-data analysis solutions often handle several issues, by giving the distributed environment the chance to scale out by adding more nodes to supply processing units and storage. Cassandra, HBase, HDFS and MongoDB are examples of large-data storage platforms that leverage shared-nothing designs to address storage limits by horizontally expanding out to new nodes, allowing for huge data expansion. The following are some of the characterized criteria used to compare the aforementioned big-data storage systems: server operating systems, methods of partitioning, data scheme, concurrency, programming languages and others, as seen in Table 3. The following are the 3 kinds of data models for storage that can be broadly grouped: (I) A file system such as HDFS. Data is saved schemaless in HDFS and taken logically at processing time based on the processing application's requirements, a technique refers to as "Schema-on-Reading". (II) Document-based, example, MongoDB; and (III) Column-based schema, example, Cassandra and Hbase [31].

Table 3. Storage technologies for big data.

Features	**Cassandra**	**Hadoop Hive**	**MongoDB**	**HBase**
OS server	FreeBSD, Linux, OS X, Windows	All operating systems that have a "Java virtual	Linux, OS X, Windows, Solaris,	Windows, Unix, Linux,
Model for storing data	Column-based	File-system	Document-based	Column based
key-value for MapReduce	Yes	Yes	Yes	Yes
Concurrency	Yes	Yes	Yes	Yes
Capabilities for in-memory	Yes	N/A	Yes	Yes
Theorem of CAP	Consistency Partition tolerance	Consistency Partition tolerance	Availability Partition tolerance	Consistency Availability
Programming languages supported	"C#, C++, Clojure, Erlang, Go, Haskell, Java, Node.js, Perl, PHP, Python, Ruby, Scala"	C++ Java PHP Python	"C#, C++, Clojure, Erlang, Go, Haskell, Java, Node.js, Perl, PHP, Python, Ruby, Scala"	"C, C#, C++, Groovy, Java, PHP, Python, Scala"
Concept consistency	Eventual Consistency Immediate Consistency	Eventual Consistency	Eventual Consistency Immediate Consistency	Immediate Consistency

Methods of APIs and other access	ODBC and JDBC	JDBC, ODBC, Thrift	Proprietary protocol using	ODBC and JDBC
Description	Large volumes of structured data can be managed with a distributed database.	Data warehouse software for querying and managing large distributed datasets, based on	One of the most popular document storage options	Open-source, networked, versioned and column-oriented database
Partitioning methods	Key partitioning	Shading	Shading	Key partitioning
Data scheme	Relational DBMS uses Amazon DynamoDB	Relational DBMS Schema-on-Reading	Schema-free	Relational DBMS uses Google Bigtable
Replication	Masterless-ring	Selectable replication factor	Master/slave Replication	Master/slave Replication
Base code	Java	Java	C++	Java

In-memory data processing has recently gained popularity in developing technologies, with RAM and flash memory replacing slower drives. As a result, we may differentiate large-data storage solutions based on their ability to handle data in memory, which is especially important for essential real-time applications. Representatives of this mechanism include MongoDB, Cassandra and HBase. As a result, in our research, we adopted Cassandra because of its superior query performance and always-on features, as well as its distributed capability for real-time applications. Cassandra has a masterless "ring" architecture, which has several advantages over traditional master-slave topologies. As a result, each node in a cluster is regarded evenly, so quorum can be achieved by using a majority of nodes.

6. REAL-TIME PROCESSING OF STATE-OF-THE-ART ARCHITECTURE

Lambda and Kappa are two real-time processing architectures that are presented in this study. We evaluated them using their specifications and came up with a stronger solution that meets the real-time requirements specified previously.

Batch processing, as shown in the literature, performs processing on huge datasets with great throughput and efficiency, but it usually takes a long time. It could take several hours, which is far too much latency for almost any current application to provide live results. Stream processing, on the other hand, works with the most recent records that enter the system, allowing for quick processing and near-real-time results, but at the cost of being less precise than batch processing. Nathan Marz proposed the Lambda architecture [40], which combines both types of processing to gain their benefits in one architecture, providing real-time results and correct perspectives with low latency and high throughput with fault tolerance. This architecture is made up of 3 levels; namely, the batch layer, the speed layer and the serving layer (depicted in Figure 8).

The batch layer generates batch views and keeps track of the master copy of the dataset. The serving layer incorporates the findings from the batch and speed layers. To compensate for the significant latency of the service layer updates, the speed layer only processes the most current data. The batch and speed processing layers are on the same level in the architecture. This means that the fresh raw data is provided to both of them at the same time. In the meantime, the serving layer is located above as seen in Figure 8. However, due to its intricacy, this architecture has significant drawbacks as well as some criticism. This architecture necessitates the integration of numerous systems and technologies, which adds to the process's complexity. In addition, because there are two processing levels, distinct processing codes must be maintained and kept in sync to provide views to the serving layer. This also highlights the fact that such routines might be written in a variety of programming languages. Finally, the serving layer is fed by two separate layers whose data, aside from the batch layer's pre-stored data, will be identical, implying that data, information and logic will be duplicated.

Kappa architecture is a lambda architecture simplification. It is a software architectural pattern designed by Jay Kreps in 2014 based on his LinkedIn experience [3]. With the exception that all data travels over a single conduit, the stream layer, the Kappa design delivers the same benefits as the Lambda architecture. Data is appended to a unified, distributed and fault-tolerant log and its status is

only updated when such appends occur. This allows for view recalculations or recomputations. To do so, the data is streamed back in from the beginning. To avoid losing the prior computation, a parallel task is started, allowing two computations to be done at the same time. Following the completion of the second computation, the developer must decide whether to keep both, combine them or remove the prior one and keep the last one if it exceeds expectations. The Kappa architecture, which is made up of two levels, is depicted in Figure 9. The results are queried using the serving layer and the stream processing jobs are executed using the stream processing layer.

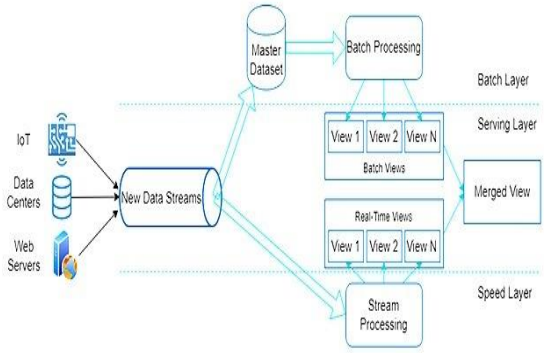


Figure 8. Lambda architecture.

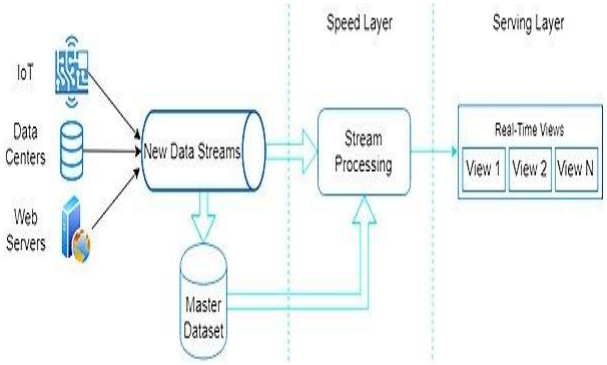


Figure 9. Kappa architecture.

Table 4 shows a brief comparison of the two architectures, Lambda and Kappa, as previously mentioned, using certain criteria.

Table 4. Comparison between lambda and kappa architecture.

Features	**Lambda**	**Kappa**
Real-time	Isn't accurate	Correct (accurate)
Fault tolerance	Yes	Yes
Architecture	Immutable	Immutable
Scalability	Yes	Yes
Permanent storage	Yes	No
Guarantees processing	Yes, in batch approximate in streaming	Exactly-once with consistency
Re-processing paradigm	During each batch cycle	Only when there is a code update
Data processing	Real-time and batch	Real-time
Layers	Batch, serving and real-time layers	Stream processing and serving layers

The lambda architecture is one of two architectures used in big-data systems and it allows for simultaneous processing of enormous datasets as well as continuous real-time access to them. The goal behind this architecture is to build two independent processes, one for batch data processing and the other for real-time data access. The batch layer performs calculations on the whole data collection. It takes time, but the data returned is complete and of good quality. The dataset in the batch layer is believed to be intact [41]. You can maintain data consistency and access to past data in this way. Incoming data is processed in real time by the real-time layer. The speed with which this layer's data may be accessed correlates to the prospect of speedier information retrieval. Unfortunately, due to a lack of historical data, not all computations can be performed [41]; hence, as seen in Table 4, real time “isn't accurate” in the lambda architecture; however, the kappa architecture can overcome this setback, providing accurate real time processing. Both architectures are fault-tolerant, immutable, scalable and have guaranteed processing ability, as can be seen in Table 4. The requirement to sustain two distinct applications: one for supporting the batch layer and the other for supporting the real-time layer, is the biggest and most frequently noted downside of the lambda architecture. Because the tools used in each layer differ, it's difficult to pick one that can serve two objectives. Unfortunately, maintaining this design is more difficult and costly [41]-[42]. The kappa architecture maintains a single pipeline; hence, it is easier to manage. Therefore, in our study, we adopted the kappa architecture as a result of its numerous advantages over the lambda architecture. However, the kappa architecture had no permanent storage, as seen in Table 4, but in our proposed framework, we introduced the permanent-storage layer.

7. PROPOSED FRAMEWORK

Taking into consideration the unique peculiarities of spatiotemporal big data, as well as overviews of stream-processing platforms, distributed message queuing systems and big-data storage technologies, the two state-of-the-art real-time architectures presented in this study have numerous advantages and disadvantages. Based on the findings of the literature, we suggested the open-source framework depicted in Figure 10, which has a unique set of properties, the most notable of which is its capacity to analyze massive amounts of spatiotemporal data in real time at high speed. It also allows an unlimited number of users to create new and unique features as well as make various reforms. The proposed framework closely resembles the kappa architecture, which provides more benefits than the lambda design. However, because the kappa architecture lacks a storage layer, we included one in our proposed framework.

In our proposed framework, there is the data source, from which is where the spatiotemporal datasets are obtained. Sensor networks, online traffic, social media, video streams and other sources could all yield distinct dataset structures, resulting in a large problem known as heterogeneous data [38]. This brings in the data ingestion layer which streams data from various upstream applications and fed to real-time downstream applications using distributed queueing management technologies. In our proposed framework we adopted Kafka as a result of its numerous advantages over other message-queueing systems. Kafka is highly scalable and most importantly can handle the challenge of heterogeneous data which is also a major problem with spatiotemporal datasets. Kafka is a data-transit technology rather than a data processing system. This distinguishes it from the competition in stream processing with high throughputs. The spatiotemporal dataset which is produced from the various data sources is transformed and filtered by Kafka and a common format is produced for either storage or immediate computations as proposed by our framework. We must first install the Kafka cluster, then launch Zookeeper and the Kafka server to get Kafka up and running. Zookeeper monitors the state of Kafka cluster nodes and keeps track of Kafka topics, partitions and other data. Kafka provides an inbuilt *KafkaProducer<k, v>* class that uses the serialization process to store streaming data in a user-defined format (e.g. *CustomObject*). It is the conversion of a specified data type into byte format [42]. The configuration properties file is used to create the Kafka producer. The topic name and CustomObject are the key-value pair. The syntax is: "*Producer<String, CustomObject> producer =new KafkaProducer<String, CustomObject> (configProperties);*". The *KafkaConsumer<k, v>* class reads and deserializes the streaming data from the Kafka producer. The process of transforming a byte format to the desired format is known as derealization [42]. The ingestion layer is very important in real-time spatiotemporal data analysis, as the cleansing and preprocessing of data is carried out here. The next layer is the real-time processing spatiotemporal data layer, which is focused on real time data processing with low latency. In our proposed framework, Flink was adopted as a result of its true real-time processing ability. Hence, our major goal is to propose a true real-time processing framework as the spatiotemporal datasets are fed into the system for prompt and immediate results. Flink has a very strong unique feature that makes it tall among other stream-processing engines or computational engines, which is CEP [37]. CEP systems assess queries against uninterrupted streams of events to find trends [29]. CEP's goal is to analyze data as it enters our system, so we don't have to keep it somewhere unless it's necessary. Also, the goal of CEP is to analyze and react to streams of events. Machine learning which is also part of our proposed framework at the processing layer is responsible for real-time prediction. Our framework has been designed to incorporate all these functionalities. At the end of the real-time processing layer, the output is sent to the storage layer, where we adopted the Cassandra as a result of its distributed ability. Cassandra has many benefits, such as a completely decentralized design with no single point of failure, promising linear scalability, great write performance and configurable data consistency levels within queries. A ring of nodes organizes the Cassandra cluster. Each of these nodes is in charge of storing a portion of the data. The hash keyspace is divided by the total number of *tokens* selected for the database to provide an equal data distribution inside the ring. A random subset of potential primary hash key values is connected with a node based on the number of *tokens* issued to it. This subset of data becomes the responsibility of the node for the entire database. Each node in the ring usually has the same quantity of *tokens*. Cassandra replicates data on other nodes in the ring to ensure high availability. The *Replication Factor (RF)* determines the number of replicas. This means that each node in a cluster of *N* nodes will store a piece of the keyspace equal to *RF=N* [43]. The visualization layer's primary responsibility is to transmit the final

data and outcomes in streaming mode to the user. If all processes are completed correctly, this layer can respond quickly.

7.1 Comparison of the Proposed Framework

The proposed framework has been designed to tackle various issues with both lambda and kappa systems. Lambda enables clients to have the most up-to-date vision. However, business logic is performed at both layer levels, two distinct sources of the same data are required to feed the next layer and this design requires many frameworks to set up. Kappa architecture was established as a result of the complexities of lambda architecture. Unlike lambda, kappa evolves to be more focused on data processing, even though it does not support permanent data storage. This architecture is less complicated than lambda and allows the user to select which implementation composers to use. However, kappa is not a magical formula that can solve all of the big spatiotemporal-data problems. Furthermore, instead of addressing data-quality issues or data-analysis outcomes, these two architectures focus on balancing throughput and latency to handle performance challenges. The kappa architectural principle underpins our proposed framework. It's a streaming data-processing approach that allows for long-term data storage by treating all incoming data as streaming data. The suggested framework can deliver actual real-time processing using Flink and machine learning. Flink is a fault-tolerant distributed real-time computing system with many other advantages, as detailed in the previous sections. By efficiently combining and expanding sophisticated real-time computations in a computer cluster, Flink enables the reliable processing of infinite streams of data. In another comparison, the proposed framework was also created to address specific aspects that are strongly linked to spatiotemporal big data in stream processing. Some of these characteristics are scalability (Sc), data analytics (DA), multiple event types (MET), prediction tools (PT), data storage (DS), real-time (Rt), performance evaluation (PE) and stream processing (SP). Table 5 compares our novel framework with others and our framework has all the capabilities, which are challenging to stream processing for big spatiotemporal data; hence, our proposed framework can handle all these characteristics.

From Table 5, the check-marks (√) indicated that the existing framework from previous works can perform the important functions (Sc, SP, DA, MET, PT, Rt and PE) as regards big spatiotemporal data in stream-processing frameworks, while the check-marks (-) indicate that the framework cannot perform the earlier listed functions, since the authors did not incorporate them in their frameworks. As discussed earlier in Section 2, these characteristics or parameters are very important for real-time stream processing with big spatiotemporal data, making the system more robust and relevant, since it has all the required characteristics; hence, in our proposed framework, we made the provisions to accommodate all these characteristics.

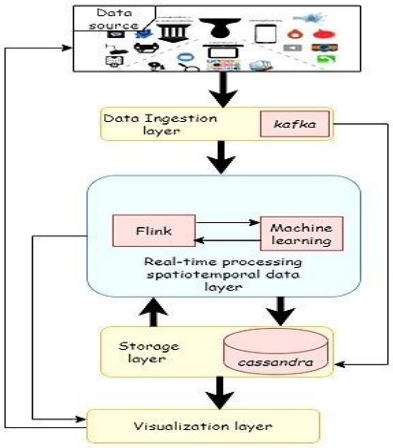


Figure 10. Proposed framework.

Table 5. Comparison of our proposed framework with others.

	Sc	SP	DA	DS	MET	PT	Rt	PE
Corral-Plaza et al., [38]	√	√	√	√	√	-	√	√
Carcillo et al., [33]	√	√	√	√	-	√	-	-
Amini et al., [44]	√	√	√	√	-	√	√	-
D'silva et al., [45]	√	√	√	√	√	-	√	√
Jung et al., [46]	√	-	√	-	-	-	-	√
Montori et al., [47]	-	-	√	√	-	-	-	-
Santos et al., [48]	-	-	√	√	-	√	√	-
Our proposal	√	√	√	√	√	√	√	√

8. CONCLUSIONS

We have developed a novel framework in this study that may be used in a variety of spatiotemporal big-data scenarios. Its key innovative advantage is the capacity to automatically handle and analyze spatiotemporal data regardless of structure. The inclusion and utilization of (1) Kafka as process

streams of spatiotemporal data sources as they occur; (2) Apache Flink as the computational layer and (3) Apache Cassandra as the storage layer for real-time distributed storage have benefited this framework. The study's major purpose is to present a true real-time processing paradigm using Flink and machine learning. In our proposed design, we suggested emphasizing the real-time processing layer and we did our best to optimize it with Flink and machine learning. The advantages of the technologies employed, as well as the advantages of kappa design after it was compared with the lambda architecture, where the key inspirations for this innovative building were obtained, are presented. The study highlighted stream-processing technologies, queueing-messaging systems and big-data storage technologies and presented their comparison for a better choice. Looking at the best tools and their advantages over others, the study proposed a novel true real time spatiotemporal data stream-processing framework. Hence, an important framework for processing and analysing spatiotemporal data from multiple sources with irregular shapes has been proposed, so that researchers can focus on data analysis instead of worrying about the data sources' structure. The following stage is to validate and assess its performance. The vast majority of research, including this one, has certain limitations. However, until the validation process is completed, we won't be able to examine its shortcomings.

ACKNOWLEDGEMENTS

The authors would like to thank everyone, just everyone!

REFERENCES

[1] B. R. Hiraman, M. C. Viresh and C. K. Abhijeet, "A Study of Apache Kafka in Big Data Stream Processing," Proc. of the Int. Conf. on Information, Communication, Engineering and Technology (ICICET 2018), pp. 1–3, DOI: 10.1109/ICICET.2018.8533771, 2018.

[2] J. Manyika, M. Chui Brown, B. B. J., R. Dobbs, C. Roxburgh and A. Hung Byers, "Big Data: The Next Frontier for Innovation, Competition and Productivity," McKinsey Global Institute, no. June, p. 156, [Online], Available: https://bigdatawg.nist.gov/pdf/MGI_big_data_full_report.pdf, 2011.

[3] S. Ounacer, M. Amine, S. Ardchir, A. Daif and M. Azouazi, "A New Architecture for Real Time Data Stream Processing," International Journal of Advanced Computer Science and Applications, vol. 8, no. 11, pp. 44–51, DOI: 10.14569/ijacsa.2017.081106, 2017.

[4] F. Pivec, "The Global Information Technology Report 2003–2004," Organizacija Znanja, vol. 8, no. 4, pp. 203-206, DOI:10.3359/oz0304203, 2003.

[5] S. Nadal et al., "A Software Reference Architecture for Semantic-aware Big Data Systems," Information and Software Technology, vol. 90, pp. 75–92, DOI: 10.1016/j.infsof.2017.06.001, 2017.

[6] A. Hamdi, K. Shaban, A. Erradi et al., "Spatiotemporal Data Mining: A Survey on Challenges and Open Problems," Artificial Intelligence Review, no. 0123456789, DOI: 10.48550/arXiv.2103.17128, 2021.

[7] N. Khan, et al., "The 10 Vs, Issues and Challenges of Big Data," Proc. of the ACM Int. Conf., no. March, pp. 52–56, DOI: 10.1145/3206157.3206166, 2018.

[8] R. L. Villars, C. W. Olofson and M. Eastwood, "Big Data: What It is and Why You Should Care," IDC White Paper, pp. 7–8, 2011, [Online], Available: http://www.tracemyflows.com/ uploads/big_data/IDC_ AMD_Big_Data_Whitepaper.pdf, 2011.

[9] N. Elgendy and A. Elragal, "Big Data Analytics: A Literature Review," Journal of Management Analytics, vol. 2, no. 3, pp. 214–227, 2014.

[10] C. L. Philip Chen and C. Y. Zhang, "Data-intensive Applications, Challenges, Techniques and Technologies: A Survey on Big Data," Information Sciences, vol. 275, pp. 314–347, 2014.

[11] S. Salehian and Y. Yan, "Comparison of Spark Resource Managers and Distributed File Systems," Proc. of the IEEE Int. Conf. on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom) (BDCloud-SocialCom-SustainCom), pp. 567–572, DOI: 10.1109/BDCloud-SocialCom-SustainCom.2016.88, 2016.

[12] J. Jo and K. W. Lee, "MapReduce-based D-ELT Framework to Address the Challenges of Geospatial Big Data," ISPRS Int. Journal of Geo-information, vol. 8, no. 11, DOI: 10.3390/ijgi8110475, 2019.

[13] J. Kang, L. Fang, S. Li and X. Wang, "Parallel Cellular Automata Markov Model for Land Use Change Prediction over MapReduce Framework," ISPRS Int. Journal of Geo-Information, vol. 8, no. 10, DOI: 10.3390/ijgi8100454, 2019.

[14] D. Glushkova, P. Jovanovic and A. Abelló, "MapReduce Performance Model for Hadoop 2.x," Information Systems, vol. 79, pp. 32–43, DOI: 10.1016/j.is.2017.11.006, 2019.

[15] I. A. T. Hashem et al., "MapReduce Scheduling Algorithms: A Review," The Journal of Supercomputing, vol. 76, pp. 4915–4945, 2020.

[16] F. Li, J. Chen and Z. Wang, "Wireless MapReduce Distributed Computing, " IEEE Transactions on

Information Theory, vol. 65, no. 10, pp. 6101–6114, DOI: 10.1109/TIT.2019.2924621, 2019.

[17] M. Bendre and R. Manthalkar, "Time Series Decomposition and Predictive Analytics Using MapReduce Framework," Expert Systems with Applications, vol. 116, pp. 108–120, 2019.

[18] S. Heidari, M. Alborzi, R. Radfar et al., "Big Data Clustering with Varied Density Based on MapReduce," Journal of Big Data, vol. 6, no. 1, DOI: 10.1186/s40537-019-0236-x, 2019.

[19] N. Maleki, A. M. Rahmani and M. Conti, "MapReduce: An Infrastructure Review and Research Insights," The Journal of Supercomputing, vol. 75, pp. 6934–7002, 2019.

[20] S. Wang, Y. Zhong and E. Wang, "An Integrated GIS Platform Architecture for Spatiotemporal Big Data," Future Generation Comp. Sys., vol. 94, pp. 160–172, DOI: 10.1016/j.future.2018.10.034, 2019.

[21] M. M. Alam, L. Torgo and A. Bifet, "A Survey on Spatio-temporal Data Analytics Systems," ACM Computing Surveys, pp. 1–37, DOI: 10.1145/3507904, 2022.

[22] Apache, "Apache Hadoop: An Open-source Distributed Processing Framework," [Online], Available: https://hadoop.apache.org/, 2020.

[23] A. Davoudian, L. Chen and M. Liu, "A Survey on NoSQL Stores," ACM Computing Surveys, vol. 51, no. 2, DOI: 10.1145/3158661, 2018.

[24] M. Zaharia et al., "Apache Spark: A Unified Engine for Big Data Processing," Communications of the ACM, vol. 59, no. 11, pp. 56–65, DOI: 10.1145/2934664, 2016.

[25] I. Yaqoob, I. A. T. Hashem, A. Ahmed, S. M. A. Kazmi and C. S. Hong, "Internet of Things Forensics: Recent Advances, Taxonomy, Requirements and Open Challenges," Future Generation Computer Systems, vol. 92, no. May 2018, pp. 265–275, DOI: 10.1016/j.future.2018.09.058, 2019.

[26] E. Ahmed et al., "The Role of Big Data Analytics in Internet of Things," Computer Networks, vol. 129, pp. 459–471, DOI: 10.1016/j.comnet.2017.06.013, 2017.

[27] S. Henning and W. Hasselbring, "How to Measure Scalability of Distributed Stream Processing Engines?" Proc. of Companion of the ACM/SPEC Int. Conf. on Performance Engineering (ICPE 2021), pp. 85–88, DOI: 10.1145/3447545.3451190, 2021.

[28] K. Kallas, F. Niksic, C. Stanford and R. Alur, "Stream Processing with Dependency-guided Synchronization," Proc. of the 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP '22), pp. 1-16, DOI: 10.1145/3503221.3508413, Seoul, Republic of Korea, 2022.

[29] N. Giatrakos, E. Alevizos, A. Artikis, A. Deligiannakis and M. Garofalakis, "Complex Event Recognition in the Big Data Era: A Survey," VLDB Journal, vol. 29, no. 1, pp. 313–352, 2020.

[30] T. Li, Z. Xu, J. Tang and Y. Wang, "Model-free Control for Distributed Stream Data Processing Using Deep Reinforcement Learning," Proc. of the VLDB Endowment, vol. 11, no. 6, pp. 705–718, 2018.

[31] R. Sahal, J. G. Breslin and M. I. Ali, "Big Data and Stream Processing Platforms for Industry 4.0 Requirements Mapping for a Predictive Maintenance Use Case," Journal of Manufacturing Systems, vol. 54, no. November 2019, pp. 138–151, DOI: 10.1016/j.jmsy.2019.11.004, 2020,

[32] D. P. Carazo, Evaluation and Deployment of Big Data Technologies on a NIDS Evaluación y Despliegue de Tecnologías Big Data Sobre un NIDS, M.Sc. Thesis, Master in Data Science, Universidad Internacional Menendez Pelayo, 2019.

[33] F. Carcillo, A. Dal Pozzolo, Y. A. Le Borgne, O. Caelen, Y. Mazzer and G. Bontempi, "SCARFF: A Scalable Framework for Streaming Credit Card Fraud Detection with Spark," Information Fusion, vol. 41, pp. 182–194, DOI: 10.1016/j.inffus.2017.09.005, 2018.

[34] H. Herodotou, Y. Chen and J. Lu, "A Survey on Automatic Parameter Tuning for Big Data Processing Systems," ACM Computing Surveys, vol. 53, no. 2, DOI: 10.1145/3381027, 2020.

[35] M. Dias de Assunção, A. da Silva Veith and R. Buyya, "Distributed Data Stream Processing and Edge Computing: A Survey on Resource Elasticity and Future Directions," Journal of Network and Computer Applications, vol. 103, no. July 2017, pp. 1–17, DOI: 10.1016/j.jnca.2017.12.001, 2018.

[36] A. Batyuk and V. Voityshyn, "Apache Storm Based on Topology for Real-time Processing of Streaming Data from Social Networks," Proc. of the 1st IEEE Int. Conf. on Data Stream Mining and Processing (DSMP 2016), no. August, pp. 345–349, DOI: 10.1109/DSMP.2016.7583573, 2016.

[37] B. Zhao, H. Van Der Aa, T. T. Nguyen, Q. V. H. Nguyen and M. Weidlich, "EIRES: Efficient Integration of Remote Data in Event Stream Processing," Proc. of the ACM SIGMOD Int. Conf. on Management of Data, no. i, pp. 2128–2141, DOI: 10.1145/3448016.3457304, 2021.

[38] D. Corral-Plaza, I. Medina-Bulo, G. Ortiz and J. Boubeta-Puig, "A Stream Processing Architecture for Heterogeneous Data Sources in the Internet of Things," Computer Standards and Interfaces, vol. 70, no. June 2019, p. 103426, DOI: 10.1016/j.csi.2020.103426, , 2020.

[39] N. Tantalaki, S. Souravlas and M. Roumeliotis, "A Review on Big Data Real-time Stream Processing and Its Scheduling Techniques," International Journal of Parallel, Emergent and Distributed Systems, vol. 35, no. 5, pp. 571–601, DOI: 10.1080/17445760.2019.1585848, 2020.

[40] N. Marz, Big Data: Principles and Best Practices of Scalable Realtime Data Systems, ISBN:978-1-61729-034-3, [S.l.]: O'Reilly Media, 2013.

[41] J. Bobulski and M. Kubanek, "Data Model for Bigdata System for Multimedia," Proc. of the ACM Int. Conf. Proceeding Series, vol. PartF16898, pp. 12–17, DOI: 10.1145/3449365.3449368, 2021.

[42] A. Bandi and J. A. Hurtado, "Big Data Streaming Architecture for Edge Computing Using Kafka and Rockset," Proc. of the 5th Int. Conf. on Computing Methodologies and Communication (ICCMC 2021), no. Iccmc, pp. 323–329, DOI: 10.1109/ICCMC51019.2021.9418466, 2021.

[43] S. Dipietro, G. Casale and G. Serazzi, "A Queueing Network Model for Performance Prediction of Apache Cassandra," Proc. of the 10th EAI Int. Conf. on Performance Evaluation Methodologies and Tools (ValueTools 2016), pp. 186–193, DOI: 10.4108/eai.25-10-2016.2266606, 2017.

[44] S. Amini, I. Gerostathopoulos and C. Prehofer, "Big Data Analytics Architecture for Real-time Traffic Control," Proc. of the 5th IEEE Int. Conf. on Models and Technologies for Intelligent Transportation Systems (MT-ITS), pp. 710–715, DOI: 10.1109/MTITS.2017.8005605, 2017.

[45] G. M. D'silva, A. Khan, Gaurav and S. Bari, "Real-time Processing of IoT Events with Historic Data Using Apache Kafka and Apache Spark with Dashing Framework," Proc. of the 2nd IEEE Int. Conf. on Recent Trends in Electronics, Information Communication Technology (RTEICT), pp. 1804–1809, DOI: 10.1109/RTEICT.2017.8256910, 2017.

[46] H. S. Jung, C. S. Yoon, Y. W. Lee, J. W. Park and C. H. Yun, "Cloud Computing Platform Based Real-time Processing for Stream Reasoning," Proc. of the 6th Int. Conf. on Future Generation Communication Technologies (FGCT 2017), pp. 37–41, DOI: 10.1109/FGCT.2017.8103400, 2017.

[47] F. Montori, L. Bedogni and L. Bononi, "A Collaborative Internet of Things Architecture for Smart Cities and Environmental Monitoring," IEEE Internet of Things Journal, vol. 5, no. 2, pp. 592–605, DOI: 10.1109/JIOT.2017.2720855, 2018.

[48] P. M. Santos et al., "PortoLivingLab: An IoT-based Sensing Platform for Smart Cities," IEEE Internet of Things Journal, vol. 5, no. 2, pp. 523–532, DOI: 10.1109/JIOT.2018.2791522, 2018.

ملخص البحث:

تُعدّ القدرة على تفسير سيول البيانات المؤقّتة حيّزيّاً أمراً حاسماً بالنسبة للعديد من الأنظمة. ومع ذلك، فإنّ معالجة كمّيات هائلة من البيانات المؤقّتة حيّزيّاً من مصادر متنوعة، مثل التّواصل عبر الإنترنت، ومنصّات التّواصل الاجتماعي، وشبكات المجسّات وغيرها، تشكّل تحدّياً ملحوظاً. لذلك، فإنّ الهدف الأساسي من هذا البحث هو إيجاد إطارٍ لمعالجة وتحليل البيانات المؤقّتة حيّزيّاً من مصادر متعدّدة وبأشكالٍ غير منتظمة، بحيث يتمكّن الباحثون من التّركيز على تحليل البيانات بدلاً من الاهتمام ببنية مصادر البيانات.

نقترح في هذا البحث نموذجاً جديداً لمعالجة سيول البيانات المؤقّتة حيّزيّاً، يمكّن من معالجة البيانات بسرعةٍ عالية وبتأخُّر طفيفٍ عن الزّمن الحقيقي، مع أخذ الاعتبارات آنفة الذّكر بعين الاعتبار. كذلك نجري مقارنةً بين النّموذج المقترح وعددٍ من النّماذج المتبنّاة في دراسات سابقة تتعلّق بمعالجة البيانات الضّخمة في الزّمن الحقيقي. هذا الى جانب نظرة شاملة على تقنيات المصادر المفتوحة المستخدمة في معالجة سيول البيانات في الزّمن الحقيقي.

يتميّز النموذج المقترح بأنّه يدمج بين (أباتشي كافكا) لاستقبال البيانات من مصادرها، و (أباتشي فلينك) لمعالجة سيول البيانات، وتعلُّم الآلة للتّوقّعات في الزّمن الحقيقي، و (أباتشي كاساندْرا) في طبقة التّخزين من أجل التّخزين الموزّع في الزّمن الحقيقي. وقد تمّت مقارنة النّموذج المقترح مع عددٍ من النّماذج الأخرى المستخدمة لمعالجة سيول البيانات في الزّمن الحقيقي، وذلك بناءً على مجموعة من الخصائص، مثل [إمكانية التّوسيع، وأدوات التّوقُّع، وتحليل البيانات، وأنواع الأحداث المتعدّدة، وتخزين البيانات، والزّمن الحقيقي، وتقييم الأداء في معالجة سيول البيانات]. وقد أثبت النّموذج المقترح تفوُّقاً على النّماذج الأخرى وبرهن على فعاليته في التّعامل مع جميع المسائل ذات العلاقة.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 08, No. 03, September 2022.

SENTIMENT ANALYSIS BASED ON PROBABILISTIC CLASSIFIER TECHNIQUES IN VARIOUS INDONESIAN REVIEW DATA

Nur Hayatin[1], Suraya Alias[2], Lai Po Hung[2], Mohd Shamrie Sainin[2]

(Received: 10-Mar.-2022, Revised: 28-Apr.-2022, Accepted: 24-May-2022)

ABSTRACT

Sentiment analysis is the field in data science to achieve a broader holistic view of users' needs and expectations. Indonesian user opinions have the potential to manage to be valuable information using sentiment-analysis tasks. One of the most supervised-learning techniques used in Indonesian sentiment analysis is the Naïve Bayes classifier. The classifier can be optimized and tuned in various models to increase the sentiment analysis model performance. This research aims to examine the performance of various Naïve Bayes models in sentiment analysis, especially when implemented in small datasets to handle overfitting problems. Four different Naïve Bayes models used are Gaussian, Multinomial, Complement and Bernoulli. We also analyze the effect of various pre-processing techniques on the models' performance. Moreover, we build the first fashion dataset from the Indonesian marketplace which has a unique character compared to the datasets from other domains. Finally, we also use various datasets in the experiment to test the Naïve Bayes models' performance. From the experimental results, Complement Naïve Bayes is superior to other models, especially in handling overfitting with an F1-score of approximately 0.82.

KEYWORDS

Naïve Bayes model, Probabilistic classifier, Sentiment analysis, Supervised learning.

1. INTRODUCTION

In Natural Language Processing (NLP), the data is dominated by text that can come from a webpage, social media, online reviews, online news, etc. Sentiment analysis is a field that can achieve a broader holistic view of customers' needs and expectations [1]. Research in this field is not only studied in English, but also in various languages, such as Malay [2], Arabic [3]-[4], as well as Indonesian. It is an important method for social sciences, because of that it is used in various disciplines including analyzing reviews from e-commerce.

Indonesia has a big consumer base of e-commerce consisting of over 8 hundred million visitors in 2019 [5]. This is a potential to identify that Indonesian user opinions from product reviews on the internet become valuable information using sentiment-analysis tasks. Studies on Indonesian sentiment analysis have grown in recent years. We have reviewed more than 100 references related to Indonesian sentiment analysis using Machine learning (ML) techniques. From the review study results, we found some popular ML techniques implemented in Indonesian research; namely, Naïve Bayes, Support Vector Machine, Decision Tree and K-Nearest Neighbour. To the best of our knowledge, there are three various Naïve Bayes models implemented in Indonesian sentiment analysis; namely, Gaussian, Multinomial [6]-[7] and Bernoulli [8]. However, the simple probabilistic classifier is the most popular technique in Indonesian sentiment-analysis research [9]-[10]. Only one research used Bernoulli Naïve Bayes [8], while Complement Naïve Bayes has not been implemented in Indonesian sentiment-analysis research. Naïve Bayes has various probabilistic models; namely, Gaussian, Multinomial, Complement and Bernoulli that can be implemented and can increase the sentiment-analysis model performance.

Some previous research in Indonesian sentiment analysis using a Naïve Bayes classifier was conducted. Priadana & Rizal developed a sentiment-analysis model based on lexicon-based and Naive Bayes Classifiers [6]. The model is used to track trending topics and analyze the sentiment of public opinion on Instagram to figure out government performance in tourism from Instagram during the COVID-19 pandemic. They also implemented some pre-processing techniques, such as lowercase, removing

1. N. Hayatin is with Informatics Department, University of Muhammadiyah Malang, Indonesia. She is also a PhD student at Faculty of Computing and Informatics, Universiti Malaysia Sabah. Email: noorhayatin@umm.ac.id
2. S. Alias (corresponding author), L.P.Hung and M.S.Saini are with Computing and Informatics Faculty, Universiti Malaysia Sabah, Kota Kinabalu, Malaysia. Emails: suealias@ums.edu.my, laipohung@ums.edu.my and shamrie@ums.edu.my

symbol, stemming, tokenizing and bag of words. Sutabri et al. [7] applied multinomial Naïve Bayes to analyze sentiment in Indonesian popular e-travelling sites. Meanwhile, other research applied a similar Naïve Bayes model to the education domain. Akbar et al. proposed a sentiment-analysis model using Bernoulli Naïve Bayes their model can differentiate between pro and contra tweets on the lockdown policy topics using Indonesian tweets [8].

This research focuses on analyzing sentiment in the Indonesian fashion dataset using the probabilistic classifier. The objective statement of the research is as follows:

1) Building a new sentiment dataset in the fashion domain from the Indonesian marketplace.
2) Examining the performance of various Naïve Bayes models in sentiment analysis for small datasets and overfitting handling.
3) Analyzing the effect of various pre-processing techniques on the model performance.

2. RESEARCH METHOD

The research methodology to conduct the research has five steps; namely, data gathering, pre-processing, feature extraction, sentiment classification and evaluation. Figure 1 depicts the methodology architecture of the research.

2.1 Data Gathering

There are various domains of data used in Indonesian sentiment analysis (see Figure 2). To the best of our knowledge, the Indonesian sentiment-analysis dataset in fashion domain has not been created before. We are the pioneers in building the dataset in this domain. From analyzing the dataset, we note some unique keywords of reviews in the Indonesian marketplace, especially in fashion opinions that are different from those in other domains. Those keywords are "*bahan*" (material), "*ukuran*" (size), "*pengiriman*" (delivery service), "*warna"* (colour) and "*harga"* (price). In this chapter, we explain the process of gathering the data.

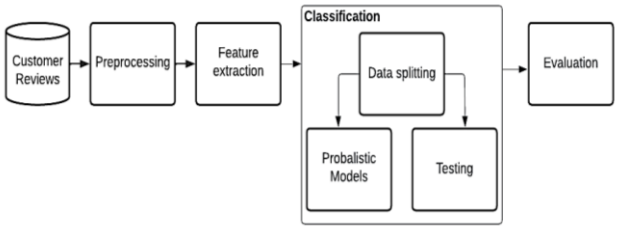


Figure 1. Methodology architecture.

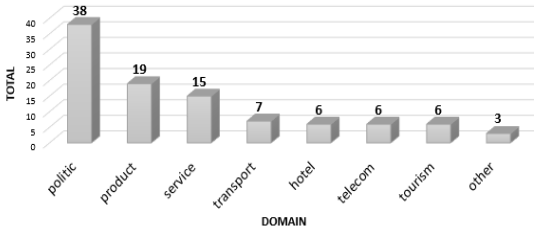


Figure 2. various domains used in Indonesian sentiment analysis.

We collect the data from product reviews scrapped from Shopee Marketplace, one of the big marketplaces in Indonesia [11]. We choose a product-related fashion. There are some criteria for products to be selected; i.e.: Total review is more than 1k comments; there is no far gap of total comments for each rating (1–5).

In gathering the data, the first step is copying the URL of the product which was selected, then scraping the product reviews using Python and Shopee API. The function used to access URL is *requests.get* that returned an object which is the HTML text; furthermore, parsing the HTML with *beautifulsoup* library to extract the HTML elements that are required.

The total data which has been scraped is 3020 reviews. This is the maximum number of data that can be scraped for one store in Shopee using the API. There are three data attributes we need to scrape; i.e.: *id order, comment and star*. We need *id order* to select unique data, because based on manual checking, there is duplicate data that is submitted from a user. This might happen because of accidentally double submitted data by a user or a system error. However, for the experiment, we only used two data; namely, comment and star. The review data that has resulted from scraping is shown in Table 1.

Data Labelling: In supervised learning, the primary step conducted before classifying is labelling the data. We generate label data automatically based on the rating score from the "star" column. Rating represents the acceptance and opinion of the product. There are five categories of rating through the

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 08, No. 03, September 2022.

Table 1. Pairs of review data and star rating.

Comment	Star
Alhamdulillah terimakasih banyak semoga sukses selalu aminnnnn bagusss banget recommended seller deh semoga sukses. (In English: Alhamdulillah, thank you very much, good luck always aminnnnn, really recommended seller, good luck).	5
baguss worthit lahh buat harga segituu, pengirimannya juga lumayan cepat ga nyesel sii beli disinii. (In English: It's good, it's worth for that price, the delivery is also quite fast, I don't regret buying it here).	4
Barang bahanya agak tipis cma lumayan buat dipake sehari2 benang dan jahitanya kurng rapih. (In English: The material is a bit, thin but it's good enough for everyday use, the thread and the stitches are not neat).	3
Oversize nya kecil sekali..kecewa. (In English: The oversize is very small..disappointed).	2
Bahannya rusak gak sesuai fto kaos nya tipis banget gak ska. (In English: The material is ruined, it doesn't match the photo, the shirt is very thin, I don't like it).	1

number of stars given by the author. The range of stars is 1 to 5, where 5 is the highest star score which is interpreted positively *vice versa* 1 is the lowest star score that is interpreted negatively. The total of comments for each rating category from 1 to 5 is 177, 107, 258, 489 and 1787, respectively.

We adopt the Likert scale to convert the rating scores. The Likert scale is a bipolar scale method that measures both positive and negative responses to a statement. In sentiment analysis, data is divided into two classes based on sentiment polarity. In this research, data is labelled as positive (represented by "1") and negative (represented by "0"). A simple rule based on rating scores is used to label data automatically. The 5-star score will be transformed automatically into a positive label ("1") and others will be generated with a negative one ("0"). The data distribution after labelling is as follows: total data in label "1" is 1787, while total data in label "0" is 1031 data.

Data Balancing: Data balancing is a step that aims to achieve similarity to the total data of each label category. There are various techniques to do this process. This research tried to get balancing using minimum data standards. From Figure 3, we know that label "0" has minimum data with a total of 1031. Therefore, label "1" will be pruned, so that the total is like that of label "0". For a simple process, both labels now have similar total data: 1000.

Finally, the clean data is produced with the total data being 2000 selected rows. However, pre-processing techniques are implemented and we clean the data, especially to filter empty data after pre-processing. Therefore, the final data filtered is a total of 520 comments. Table 2 shows the transformation of the data starting from scraping to balancing.

Table 2. Total data transformation.

Original data from scraping	Drop null	Data balancing	
		Before pre-processing	After pre-processing
3020	2818	2000	520

To better understand the data, we visualize the group of data in the form of a word cloud based on a sentiment label. Word cloud contains the terms selected from the dataset and then shown based on the higher frequency of occurrence of the term. For each label, positive and negative, visualization is depicted in Figure 3. From the positive word cloud in Figure 3(a), we can see that there are some dominant words, such as *barang bagus, kiriman cepat, harga sesuai* (in English: good item, fast delivery, good price). Meanwhile, some words represent a negative expression, such as *barang tidak sesuai, size kecil, kecewa, rusak* (in English: the item does not match, the size is small, disappointed, damaged), as presented in Figure 3 (b).

(a) (b)

Figure 3. Word cloud visualization; (a) positive and (b) negative.

2.2 Pre-processing

The second step implemented in this research is pre-processing after gathering the data. There are some stages in pre-processing; to simplify, we group those into three main stages of pre-processing; namely, data cleansing, data transforming and data tagging. Figure 4 depicts the order of the main stages of pre-processing.

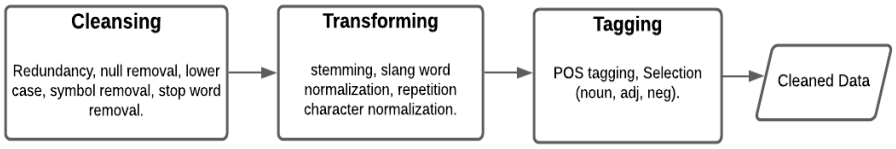


Figure 4. Pre-processing main stages.

Data Cleansing. This is the first stage of pre-processing. There are some stages to clean the data included; i.e.: redundancy, null-value removal, lower case, symbol removal and stop word removal. Redundancy removal is a process aimed to get a unique row; it was selected based on the "*id order*". After this process is carried out to the dataset, the total data is still in 3020 rows, so the data collection is unique or has no redundancy.

The next step is the null-value removal. This process is targeted to select a row that has a null value and then remove it. After it was executed to the dataset, there is a reduction of the total data of the dataset from 3020 rows to 2818 reviews. Lower case is a process to change each abjad of the sentence to be lower. It is used to reduce the data dimension; for example, there is the phrase= {"*Baju bagus", "baju bagus", "Baju Bagus*"}; if we do not use lower case, then after tokenizing these three phrases will be saved as 4-term collection= {"*Baju", "baju", "Bagus", "bagus*"}. However, if we use lower case, then there is only 2-term collection= {"*baju", "bagus*"}. It is produced, because the system will be saved for each unique word based on the character.

This research use symbol removal to clean the data. The symbols that will be removed involve punctuation, ASCII, UNICODE & Newline. This stage automatically includes emoji removal. The last stage of data cleansing is the stop-word removal step. This process is to remove unimportant words. We use a dictionary containing standard words to select stop words; if the word is not in accordance with a word in the dictionary, then that word will be removed.

Data Transforming. At this stage, the words of the sentences will be changed into different word forms. The processes in this stage are stemming, slang word normalization and repetition of character normalization. The last two processes are a part of spelling correction. A previous study has used this technique for pre-processing and the effectiveness of the model has been shown [12].

First, we use stemming to change stem words into root words. The stemming library used in this research is Sastrawi stemmed, an Indonesian stemming algorithm. The stemming result example; i.e.: stem word "*pengiriman*" (in English: delivery), will proceed to be the root word "*kirim*" (in English: send).

Many conversations on online media are done using slang words. This trend also happens in the marketplace, especially in the Indonesian customer reviews with the users' style for expressing their words. Based on the researchers' observation, the type of slang word that is usually used in the marketplace is Collegial. Colloquial is a socio-linguistic term related to a non-formal or informal language which is also referred to as a daily language [13]. The hallmark of this language, among others, is the reduced use of linguistic features, such as letters and syllables in sentences. Slang word normalization is needed to transform slang words, words that are unrecognizable in the dictionary, to be standard words. In this research, we use the slang word dictionary from Okky Ibrahim Github[1].

The next technique used for data transformation is repetition character normalization. This technique is like the previous technique and normalizes unstandardized words to be standard. Unstandardized words are related with that there is the same character mentioned in repetition in the sentence. Table 3 shows an example of a comment before and after being implemented with repetition of character normalization.

[1] https://github.com/okkyibrohim/id-abusive-language-detection/blob/3f511561df6b1ae60f7343f8992d1471209ff10b/kamusalay.csv

Table 3. Repetition character normalization.

Example: "*sumpah **iniii tokooo gercepp bangettt**, mesen kemeja **kemarennn langsungg dikirimmm** juga **hariii ituuuuu**!!! bahannya juga **tebelll pokoknyaa** tidak mengecewakan!!!! **cusss gaisss beliiii disiniiii** di jamin **bagusss**!!"*
Normal: "*sumpah **ini toko gercep banget**, mesen kemeja **kemaren langsung dikirim** juga **hari itu**!!! bahannya juga **tebel pokoknya** tidak mengecewakan!!!! **cus gais beli disini** di jamin **bagus**!!"*

Data Tagging. The final stage of pre-processing is data tagging. This stage will split the data word by word and then give the relevance tag for each word of the sentence; it is generally mentioned as POS (Part of Speech) tagging. In this research, we used *CRFTagger()* library, an Indonesian tagger. After each word is tagged, we can select which words will be used, where this research filters words classified as NN (Noun), JJ (Adjective) and Neg (Negation). This data-tagging result is also needed when we generate word cloud visualization.

2.3 Feature Extraction

We use TFIDF (Term Frequency Inverse Document Frequency) to extract the features [14]. The feature of review sentences is in the form of text. Therefore, TFIDF is needed to generate text to number through term weighting. The concept of TFIDF is that the word T_i is important if it occurs frequently. The values of the vector elements W_i for a document *d* are calculated as a combination of the statistics TF and IDF. The calculation of W_i is as follows:

$$W_i = TF(t_i, d) \cdot IDF(t_i) \tag{1}$$

where W_i is the weight of word t_i in document *d*. The term frequency *TF(t,d)* is the number of times word *t* occurs in document *d*, while the document frequency *DF(t)* is the number of documents in which the word *t* occurs at least once. The inverse document frequency *IDF(t)* can be calculated from the document frequency by:

$$IDF(t) = log\left(\frac{|D|}{DF(t)}\right) \tag{2}$$

where |*D*| is the total number of documents. The inverse document frequency of a word is low if it occurs in many documents and is highest if the word occurs in only one.

2.4 Sentiment Classification

Sentiment classification is a process to classify the data which is grouped based on the relevant sentiment class. In this research, we will classify the data into two classes based on positive sentiment and negative sentiment. We implement a classifier model; namely, Naïve Bayes.

Naive Bayes is a supervised-learning algorithm that is the simplest form of a Bayesian network [15]. This algorithm works based on Bayes' theorem with the "naive" assumption of conditional independence, where all attributes are independent given the value of the class variable. Given class variable *c* and dependent feature vector x_1 to x_n in document *d*; the probability of each sentiment class *c* is calculated as:

$$P(c, x_i) = \frac{P(x_i, c) \cdot P(c)}{P(x_i)} \tag{3}$$

where $P(x_i)$ is the same for all classes; then the class label of x_i can be determined by:

$$label(x_i) = argMax_c\{P(c, x_i)\} \tag{4}$$

There are various models of Naïve Bayes, such as Gaussian, Multinomial, Complement and Bernoulli. The difference between each Naïve Bayes model is determined by the calculation of probability $P(x_i, c)$.

Gaussian Naïve Bayes. In Gaussian NB, feature values of terms for each class *c* are usually generated by a separate Gaussian [16], where σ and μ of the feature values of words are associated with class *c*. The likelihood of feature x_i is given by:

$$P(x_i, c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} exp\left(-\frac{(d_i - \mu_c)^2}{2\sigma_c^2}\right) \tag{5}$$

The parameters σ is the variance vector and μ is the mean, while these parameters are estimated using maximum likelihood from the training document set *d*.

Multinomial Naïve Bayes. The multinomial NB classifier captures the term frequency of the documents [17]. This model is implemented for multinomially distributed data, so that it is suited for discrete feature classification.

For each class *c*, where *n* is the size of the vocabulary in all classes of the training dataset, the probability $P(x_i, c)$ of feature *i* appearing in a sample belonging to class *c* is estimated by a smoothed version of maximum likelihood as follows:

$$P(x_i, c) = \frac{N_{ci}+\alpha}{N_c+\alpha n} \quad (6)$$

where $N_{ci} = \sum_{x \in T} x_i$ is the number of times feature *i* appears in a sample of class *c* in the training set *T* and $N_c = \sum_{i=1}^{n} N_{ci}$ is the total count of all features for class *c*.

The smoothing priors $\alpha \geq 0$ account for features not present in the learning samples and prevent zero probabilities in further computations. Setting $\alpha=1$ is called Laplace smoothing, while $\alpha<1$ is called Lidstone smoothing.

Complement Naïve Bayes. This model is an adaptation of the standard multinomial Naive Bayes (MNB) algorithm that is particularly suited for imbalanced datasets. Complement NB uses statistics from the complement of each class to compute the model's weights [18]. It will lessen the bias in the weight estimates and will improve the classification accuracy. The procedure for calculating the weights is as follows:

$$P(x_i, c) = \frac{N_{\hat{c}i}+\alpha_i}{N_{\hat{c}}+\alpha} \quad (7)$$

where $N_{\hat{c}i} = \sum_{j:y_j \neq c} d_{ij}$ is the number of times word *i* occurred in documents in classes other than *c* and $N_{\hat{c}} = \sum_{j:y_j \neq c} \sum_k d_{kj}$ is the total number of word occurrences in classes other than *c* and α_i and α are smoothing hyperparameters. The classification rule is:

$$\hat{c} = \arg min_c \sum_i t_i w_{ci} \quad (8)$$

where a document is assigned to the class that is the poorest complement match.

Bernoulli Naïve Bayes. This type of classifier assumes that the features are binary and require only 2 values, where each value shows whether a word occurs or does not occur at least once in the document; with a value ranging between 0 and 1 [19]. The decision rule for Bernoulli Naive Bayes is based on:

$$P(x_i, c) = P(x_i, c) b_i + (1 - p(x_i|c))(1 - b_i) \quad (9)$$

where *x* is a word in the document; If the word x_i is present in the document, then $b_i = 1$ and the likelihood was $P(x_i, c)$. If the word x_i is absent, then $b_i = 0$ and the probability is $(1 - p(x_i|c))$.

3. RESULT AND DISCUSSION

In this research, we measure the performance of the probabilistic classifiers in the sentiment-analysis model. For the evaluation model, we start with preparing the datasets needed to build the model and to test the model performance. After that, data is cleaned using various pre-processing techniques. Finally, the data is classified with employing various Naïve Bayes models; namely, Gaussian, Multinomial, Complement and Bernoulli.

3.1 Dataset

The dataset used in the research is from both primary and secondary data. The dataset needs to be split, because we use supervised learning that needs trained data to build the model. Dataset is separated into two groups: train data and test data. In this research, data smoothing for each sentiment class is noticed to produce balanced data.

For the primary dataset, we use a fashion dataset that contained 520 reviews scraped from the Indonesian marketplace. The detailed process for scraping data is explained in Section 2. Total number of data for training is 416, while that of test data is 104. Table 4 shows the proportions of the total primary data split for each sentiment label.

For the secondary data, we gathered various Indonesian reviews as a benchmark from the open public

dataset used in sentiment-analysis research[2]. Four public datasets were utilized especially for conducting the third experiment scenario; namely, cellular [20], cyberbullying [21], movie [22] and politic [23]. The details of proportions for each dataset can be seen in Table 5.

Table 4. Primary data.

Total	Train		Test	
	Positive	Negative	Positive	Negative
520	216	200	53	51

Table 5. Various Indonesian public datasets for sentiment analysis.

Dataset	#Data	Positive	Negative
cellular	300	169	139
cyberbullying	400	200	200
movie	200	100	100
politic	900	450	450

3.2 Experimental Setup

We use Scikit Learn library to implement the algorithms of Naïve Bayes models; namely, Gaussian, Multinomial, Complement and Bernoulli [24]. We have three scenarios of the experiment. In the first scenario, we test some pre-processing techniques to analyze which pre-processing technique affects the model performance. In the second scenario, we test various Naïve Bayes models in sentiment analysis using a fashion dataset. And in the last scenario, we use some public datasets in Indonesian sentiment analysis to measure the models' performance as well as to examine which model is appropriate for handling overfitting.

Considering the amount of data which is under 1000 rows, we implement the K-fold cross-validation method to handle overfitting. We use standard K=5 in the experiment and run standard statistical tools, such as F1-score, precision, recall and accuracy to assess both training and validation performance.

Experiment #1. In the first experiment, we examine the effect of pre-processing techniques on sentiment analysis. Pre-processing is the first step in sentiment analysis or other tasks related to text analyzing. This step is important to understand the data and it was proven that it can improve the model accuracy [25]. However, all of them are not appropriate to be implemented for a small dataset, so there is a need to understand which technique is more influential in increasing the sentiment-model performance.

There are eight pre-processing techniques implemented in this experiment; namely, lower case, punctuation, number and unicode removal, stop-word removal, slang-word normalization, character-repetition normalization, stemming and POS tagging. The detailed explanation for each technique is explained in section 2. In the first step, we design some scenarios by combining some pre-processing techniques into six cases. The six combinations of pre-processing techniques are presented in Table 6.

Case 1 represents a scenario without considering pre-processing techniques; the data proceed in this scenario is from original reviews. Case 2 only uses standard pre-processing techniques, such as lower case, punctuation, symbol removal and stop-word removal. Case 3 and Case 4 implement slang-word and character-repetition handling, respectively. Meanwhile, stemming is added in Case 5; in this experiment we used a standard stemming algorithm from the Sastrawi library. Finally, a complete technique version which uses POS-tagging filtering is employed in Case 6.

Table 6. Combination of pre-processing techniques.

Case	Pre-processing Technique							
	Lower	Punct.	Symbol	Stop	Slang	Repeat	Stem	POS
1	-	-	-	-	-	-	-	-
2	v	v	v	v	-	-	-	-
3	v	v	v	v	v	-	-	-
4	v	v	v	v	v	v	-	-
5	v	v	v	v	v	v	v	-
6	v	v	v	v	v	v	v	v

We implement various Naïve Bayes models in this experiment. The results shows that Complement Naïve Bayes achieves a good performance compared to other models. Table 7 presents F1-score as well as accuracy of Complement Naïve Bayes using variation cases in the fashion dataset. The highest F1-

[2] https://github.com/rizalespe/Dataset-Sentimen-Analisis-Bahasa-Indonesia

score, as well as accuracy, are shown in Case 4, amounting to around 0.87 and 88.08%, respectively. Meanwhile, the lowest scores are shown in Case 2 (F1=0.83, accuracy=84.2%) and Case 4 (F1=0.83, accuracy=84.8%).

Table 7. Experiment results using variation cases for fashion dataset.

Pre-processing	**Case**	**Accuracy (%)**	**F1-score**
No	1	85.58	0.843
Yes	2	84.23	0.832
	3	86.54	0.855
	4	88.08	0.870
	5	87.69	0.866
	6	84.81	0.833

We also analyzed the results of all variation cases implemented for all datasets. Table 8 depicts the average F1-score for each case and Figure 5 presents the trend of the results. From the results, we can see a stable score appearing in Cases 3-5 of approximately 0.82. Meanwhile, the trend shows a significant decrease of F1-score in Case 6 of around 0.7, where this score is the lowest F1-score of all cases. Based on our analysis, the decreased performance in Case 6 is caused by the selection process of some class words based on POS tagging. The class of words that are selected in this pre-processing phase are noun (NN), adjective (JJ) and negation (NEG). This process reduced the dimensions of data and affected data for small datasets significantly.

Table 8. The results of all datasets for each case.

Dataset	**Case**					
	1	2	3	4	5	6
cellular	0,785	0,785	0,800	0,800	0,800	0,697
cyber	0,820	0,822	0,855	0,855	0,855	0,770
movie	0,838	0,838	0,852	0,852	0,852	0,707
politic	0,730	0,728	0,743	0,743	0,742	0,627
fashion	0,843	0,832	0,855	0,870	0,866	0,833
average	0,803	0,801	0,821	0,824	0,823	0,727

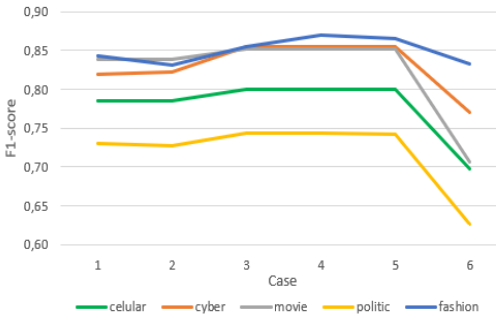


Figure 5. Average F1-score from all domain datasets for each case.

The data scraped from the marketplace causes the format to be unstructured, so pre-processing is needed to clean and prepare the data before analyzing. However, various pre-processing techniques are not appropriate to be implemented, especially for a small dataset. Slang-word and repetition handling and stemming are powerful to be employed. On the other hand, based on the experiment, the selection of words using POS tagging is not recommended for supervised learning with a small dataset, because it reduces the dimensions of data.

Experiment #2. We compare the sentiment-classification results from various types of Naïve Bayes models for the fashion dataset. The fashion dataset is a primary dataset used in the experiment (see Table 4 for the details of the primary data). Four different types of Naïve Bayes models are implemented in this experiment; namely, Gaussian, Multinomial, Complement and Bernoulli. The experiment is conducted to measure the performance of each Naive Bayes model in classifying sentiment sentences. We use the K-fold cross-validation method with K=5. K-fold cross-validation is a measurement method for both training and validation performance that is appropriate for small data. Table 9 presents the validation result for each model of Naïve Bayes in terms of F1-score, accuracy, precision and recall for the fashion dataset.

Table 9. Validation performance results of the second experiment.

Model	**F1**	**Prec**	**Rec**	**Acc(%)**
Gaussian	0.668	0.881	0.538	73.27
Multinomial	**0.876**	**0.919**	0.840	**88.18**
Complement	**0.876**	**0.919**	0.840	**88.18**
Bernoulli	0.847	0.839	**0.855**	84.55

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 08, No. 03, September 2022.

From Table 9, we can see that both Multinomial and Complement present the highest F1-score, precision and accuracy, while the highest recall is produced by Bernoulli. The highest F1-score, precision, accuracy and recall for the fashion dataset are 0.876, 0.919, 0.855 and 88.18%, respectively. Meanwhile, Gaussian has the lowest measurement results with an F1-score of around 0.668 and an accuracy of around 73.27%. The experiment results show that both Complement and Multinomial models have similar performances and are superior to other Naïve Bayes models for the fashion dataset.

In Section 2, we have explained how to gather fashion dataset from Indonesian marketplace; and in Figure 3, we present two groups of words based on sentiment labels which are visualized using cloud word. We note some unique keywords that relate to the experiment results. Some keywords, such as *material* and *size*, are usually followed by context-dependent opinions, such as *thin*, *thick*, *big* and *small*. The context-dependent opinion is an opinion which appears in several aspects with uncertain polarity [26]. The sentiment polarity of context-dependent opinions is caused by the domain of the dataset. For example, the word "*thin*" is positive when mentioned in the context of cellular or electronic products. However, this word will be negative if it appears in fashion. Context-dependent opinions can affect the sentiment-analysis task performance. Therefore, there is a potential of the existence of a concern on this issue for further study.

Experiment #3. In the final experiment, we examine the performance of various Naïve Bayes models using primary data as well as secondary data. The total dataset utilized in the third experiment is comprised of five datasets from various domains; namely, cellular, cyberbullying, movie, politic and fashion. Table 10 shows the average of each statistical measurement result from various sentiment-analysis datasets for both training and validation in the third experiment.

Table 10. Comparing the results of the models in training and validation.

Model	**Training**				**Validation**			
	F1	Prec	Rec	Acc	F1	Prec	Rec	Acc
Gaussian	**0,971**	0,991	0,960	97,53	**0,703**	0,748	0,684	72,09
Multinomial	**0,974**	0,983	0,966	97,51	**0,816**	0,870	0,782	82,79
Complement	**0,978**	0,983	0,973	97,83	**0,820**	0,851	0,798	82,59
Bernoulli	**0,964**	0,963	0,966	96,41	**0,799**	0,827	0,788	80,74

In general, the Complement model is dominant over the other models for both training and validation scoring results. In training, the highest scores for F1-score, recall and accuracy produced by the Complement model are 0.978, 0.973 and 97,83%, respectively. For precision, Gaussian gave the highest precision of around 0.991. In validation evaluation, Complement and Multinomial show excellent results compared to the other two models. The highest F1-score and recall are 0.820 and 0.798, respectively produced by the Complement model. Meanwhile, Multinomial presents higher scores for precision and accuracy of around 0.870 and 82.79% respectively.

From Table 10, we can assess which model has a good performance to handle overfitting. The consistency scores in training and validation can be an indicator of a model for overfitting issues. The Complement model shows the smallest gap in F1-score between training and validation from 0.978 to 0.820. Multinomial has a small distance from training of 0.974 to validation of 0.816. Bernoulli shows a higher F1-score in training of around 0.964, while in validation, the score is under 0.80. On the other hand, the Gaussian model presents a good performance in training above 0.90 for all measurement scores, but it produces the lowest scores under 0.75 in validation.

We compare the experiment results with the baseline. An increased F1-score of the model proposed is shown in cyberbullying dataset at around 0.856, while the baseline using Support Vector Machine produces an F1-score of 0.697 [21]. Meanwhile, the Complement Naïve Bayes for political dataset presents an increased accuracy of around 72,4% compared with the baseline of 70,2% using Multinomial Naïve Bayes [23]. For the cellular dataset, the baseline using Support Vector Machine presents a similar result to that of our model using Complement Naïve Bayes (F1-score=0.800) [20]. On the other hand, the baseline of the movie dataset that used Multinomial Naïve Bayes shows an F1 score=0.917, which is higher than that of the model proposed [22]. This inconsistent result is possibly caused because there is no consideration of cross-validation in the evaluation method of the baseline. The baseline of the movie dataset did not handle the overfitting issue in the experiment.

Referring to Table 11, the Complement model has the highest (average) F1-score at 0.820, followed by the Multinomial model at 0.816. Meanwhile, the Gaussian score is the lowest in performance beingaround 0.703. This result shows that both Complement and Multinomial have good performance in sentiment analyzing, especially to handle small datasets. On the other hand, Gaussian is not good enough to handle overfitting. In terms of *politic* dataset, this dataset is bigger than the others, but this has not increased the performance. Therefore, we can conclude that a lot of data is not enough in supervised learning, but it is important to know the variance as well as the characteristics of the data.

Table 11. Average F1-score of validation results for each NB model.

Dataset	Gaussian	Multinomial	Complement	Bernoulli
cellular	0.734	0.781	0.800	0.760
cyberbullying	0.769	0.856	0.856	0.860
movie	0.704	0.831	0.831	0.786
politic	0.643	0.737	0.737	0.740
fashion	0.668	0.876	0.876	0.847
Average	**0.703**	**0.816**	**0.820**	**0.799**

4. CONCLUSIONS

This research focuses on analyzing sentiment in the fashion domain from Indonesian review data using various Naïve Bayes models. Four different Naïve Bayes models are used in this research; namely, Gaussian, Multinomial, Complement and Bernoulli. From the experiment results, we have three findings: 1) Selection of words using POS tagging is not recommended for supervised learning with a small dataset, because it can reduce the dimensions of data. 2) Complement model is superior to other models, especially to handle overfitting. 3) There are opinion words which appear in several aspects with uncertain polarity called context-dependent opinions, which can affect the sentiment-analysis task performance. For future work, choosing a powerful stemming algorithm in pre-processing can be considered as possible to increase the model performance. Other than that, knowing data characteristics and domain is crucial. Further, it will be of importance to concern the study of context-dependent opinion issues in the next experiments.

ACKNOWLEDGEMENTS

This work is supported by Kementerian Pengajian Tinggi Malaysia, Fundamental Research Grant Scheme (FRGS) by code number FRGS/1/2020/ICT02/UMS/02/2.

REFERENCES

[1] O. Alqaryouti, N. Siyam, A. A. Monem and K. Shaalan, "Aspect-based Sentiment Analysis Using Smart Government Review Data," Applied Computing and Informatics, DOI: 10.1016/j.aci.2019.11.003, 2019.

[2] S. Ainin, A. Feizollah, N. B. Anuar and N. A. Abdullah, "Sentiment Analyzes of Multilingual Tweets on Halal Tourism," Tourism Management Perspect., vol. 34, no. Feb., p. 100658, 2020.

[3] K. M. O. Nahar, A. Jaradat, M. S. Atoum and F. Ibrahim, "Sentiment Analysis and Classification of Arab Jordanian Facebook Comments For Jordanian Telecom Companies Using Lexicon-based Approach and Machine Learning," Jordanian J. of Computers and Inf. Technol. (JJCIT), vol. 6, no. 3, pp. 247–262, 2020.

[4] H. Elfaik and E. H. Nfaoui, "Deep Bidirectional LSTM Network Learning-based Sentiment Analysis for Arabic Text," J. Intellgent Systems, vol. 30, no. 1, pp. 395–412, DOI: 10.1515/jisys-2020-0021, 2021.

[5] M. Gusti, "Ini Dia Nilai Transaksi Marketplace Indonesia 2020," Kompas TV, [Online], Available: kompas.tv/article/107064/ini-dia-nilai-transaksi-marketplace-indonesia-2020. (Accessed Jun. 02, 2021).

[6] A. Priadana and A. A. Rizal, "Sentiment Analysis on Government Performance in Tourism during the COVID-19 Pandemic Period with Lexicon Based," CAUCHY, vol. 7, no. 1, pp. 28–39, Nov. 2021.

[7] T. Sutabri, S. J. Putra, M. R. Effendi, M. N. Gunawan and D. Napitupulu, "Sentiment Analysis for Popular e-traveling Sites in Indonesia Using Naive Bayes," Proc. of the 6th Int. Conf. on Cyber and IT Service Management (CITSM), pp. 1–4, DOI: 10.1109/CITSM.2018.8674262, 2018.

[8] A. F. Akbar, A. B. Santoso, P. K. Putra and I. Budi, "A Classification Model to Identify Public Opinion on the Lockdown Policy Using Indonesian Tweets," J. Theor. Appl. Inf. Technol., vol. 99, no. 14, 2021.

[9] C. C. P. Hapsari, W. Astuti and M. D. Purbolaksono, "Naive Bayes Classifier and Word2Vec for Sentiment Analysis on Bahasa Indonesia Cosmetic Product Reviews," Proc. of the Int. Conf. on Data Science and Its Applications (ICoDSA), pp. 22–27, DOI: 10.1109/ICoDSA53588.2021.9617544, Oct. 2021.

[10] A. Rahmatulloh, R. N. Shofa, I. Darmawan and Ardiansah, "Sentiment Analysis of Ojek Online User Satisfaction Based on the Naïve Bayes and Net Brand Reputation Method," Proc. of the 9th Int. Conf. on Information and Communication Technology (ICoICT), pp. 337–341, 2021.

[11] Webretailer, "Online Marketplaces in Southeast Asia: A Unique Region for Ecommerce," 2020, [Online], Available: https://www.webretailer.com/b/online-marketplaces-southeast-asia/. (Accessed Jun. 26, 2021).

[12] U. Rhohmawati, I. Slamet and H. Pratiwi, "Sentiment Analysis Using Maximum Entropy on Application Reviews (Study Case: Shopee on Google Play)," JITEKI Journal, vol. 5, no. 1, pp. 44–49, 2019.

[13] E. Swandy, "Bahasa Gaul Remaja Dalam Media Sosial Facebook," J. Bastra, vol. 1, no. 4, pp. 1–19, 2017.

[14] L. Jing, H. Huang and H. Shi, "Improved Feature Selection Approach TFIDF in Text Mining," Proc. of the 1st IEEE Int. Conf. on Machine Learning and Cybernetics, pp. 4–5, Beijing, China, 2002.

[15] J. Chen, H. Huang, S. Tian and Y. Qu, "Feature Selection for Text Classification with Naïve Bayes," Expert Syst. Appl., vol. 36, no. 3 PART 1, pp. 5432–5435, DOI: 10.1016/j.eswa.2008.06.054, 2009.

[16] Shuo Xu, "Bayesian Naïve Bayes Classifiers to Text Classification," J. Information Science, no. 15, pp. 1–12, DOI: 10.1177/0165551510000000, 2016.

[17] D. H. Abd, A. T. Sadiq and A. R. Abbas, "Political Articles Categorization Based on Different Naïve Bayes Models," Proc. of the Int. Conf. on Applied Computing to Support Industry: Innovation and Technology (ACRIT 2019), vol. 1174, pp. 286-301, 2020.

[18] J. D. M. Rennie, L. Shih, J. Teevan and D. Karger, "Tackling the Poor Assumptions of Naive Bayes Text Classifiers," Proc. of 21st Int. Conf. on Machine Learning (ICML '04), vol. 2, no. 1973, pp. 616–623, 2003.

[19] V. Metsis, I. Androutsopoulos and G. Paliouras, "Spam Filtering with Naive Bayes - Which Naive Bayes?," Proc. of the 3rd Conf. on Email and Anti-Spam (CEAS 2006), [Online], Available: https://www2.aueb.gr/users/ion/docs/ceas2006_paper.pdf, 2006.

[20] U. Rofiqoh et al. "Analisis Sentimen Tingkat Kepuasan Pengguna Penyedia Layanan Telekomunikasi Seluler Indonesia Pada Twitter Dengan Metode Support Vector Machine dan Lexion Based Feature," J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya, vol. 1, no. 12, pp. 1725–1732, 2017.

[21] W. Athira, I. Gholissodin and R. S. Perdana, "Analisis Sentimen Cyberbullying Pada Komentar Instagram Dengan Metode Klasifikasi Support Vector Machine," J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya, vol. 2, no. 11, pp. 4704–4713, 2018.

[22] P. Antinasari, R. S. Perdana and M. A. Fauzi, "Analisis Sentimen Tentang Opini Film Pada Dokumen Twitter Berbahasa Indonesia Menggunakan Naive Bayes Dengan Perbaikan Kata Tidak Baku," J. Pengemb. Teknol. Inf. dan Ilmu Komput., vol. 1, no. 12, pp. 1718–1724, 2017.

[23] A. R. T. Lestari, R. S. Perdana and M. A. Fauzi, "Analisis Sentimen Tentang Opini Pilkada DKI 2017 Pada Dokumen Twitter Berbahasa Indonesia Menggunakan Näive Bayes dan Pembobotan Emoji," J. Pengemb. Teknol. Inf. dan Ilmu Komput., vol. 1, no. 12, pp. 1718–1724, 2017.

[24] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," J. Machine Learning Research, vol. 12, pp. 2825-2830, 2011.

[25] A. W. Pradana and M. Hayaty, "The Effect of Stemming and Removal of Stopwords on the Accuracy of Sentiment Analysis on Indonesian-language Texts," Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control, vol. 4, pp. 375–380, DOI: 10.22219/kinetik.v4i4.912, 2019.

[26] H. Kansal and D. Toshniwal, "Aspect Based Summarization of Context Dependent Opinion Words," Procedia Computer Science, vol. 35, pp. 166–175, DOI: 10.1016/j.procs.2014.08.096, 2014.

ملخص البحث:

يهدف هذا البحث الى فحص أداء نماذج مختلفة من مصنِّف (NB) في تحليل المشاعر، خصوصاً عند تطبيقها على مجموعات بيانات صغيرة؛ من أجل معالجة مشكلة فرْط المواءمة. النماذج الأربعة المستخدمة هي: النّموذج الغاوْسيّ، والنّموذج متعدّد الدّوالّ، والنّموذج المتمّم، ونموذج برنولي. كذلك تمّ تحليل أثر التّقنيات المختلفة للمعالجة القَبْليّة على أداء كلٍّ منْ تلك النّماذج. من ناحيةٍ أخرى، قُمنا ببناء مجموعة بياناتٍ للأزياء من الأسواق الإندونيسية، وهي الأولى من نوْعها في حقل الموضة. وهي متميّزة في خصائصها على شبيهاتها في الحقول الأخرى. كذلك استخدمنا عدداً من مجموعات البيانات في تجربة لفحص مصنِّفات (NB) ومقارنة أداء نماذجها المختلفة. واتّضح من النّتائج تفوُّق نموذج مصنِّف (NB) المتمّم على النّماذج الأخرى ، وبخاصّةٍ فيما يتعلّق بمعالجة مشكلة فرْط المواءمة محقّقاً درجة (F1) تصل الى (0.82) .

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 08, No. 03, September 2022.

RISK FACTOR IDENTIFICATION FOR STROKE PROGNOSIS USING MACHINE-LEARNING ALGORITHMS

Tanvir Ahammad

(Received: 16-May-2022, Revised: 6-Jul.-2022, Accepted: 7-Jul.-2022)

ABSTRACT

Stroke is a life-threatening condition causing the second-leading number of deaths worldwide. It is a challenging problem in the public-health domain of the 21st century to healthcare professionals and researchers. So, proper monitoring of stroke can prevent and reduce its severity. Risk-factor analysis is one of the promising approaches for identifying the presence of stroke disease. Numerous researches have focused on forecasting strokes in patients. The majority had a good accuracy ratio, around 90%, on the publicly available datasets. Combining several pre-processing tasks can considerably increase the quality of classifiers, an area of research need. Additionally, researchers should pinpoint the major risk factors for stroke disease and use advanced classifiers to forecast the likelihood of stroke. This article presents an enhanced approach for identifying the potential risk factors and predicting the incidence of stroke on a publicly available clinical dataset. The method considers and resolves significant gaps in previous studies. It incorporates ten classification models, including advanced boosting classifiers, to detect the presence of stroke. The performance of the classifiers is analyzed on all possible subsets of attribute/feature selections concerning five metrics to find the best-performing algorithms. The experimental results demonstrate that the proposed approach achieved the best accuracy on all feature classifications. Overall, this study's main achievement is obtaining a higher percentage (97% accuracy using boosting classifiers) of stroke prognosis than state-of-the-art approaches to stroke dataset. Hence, physicians can use gradient and ensemble boosting-tree-based models that are most suitable for predicting patients' strokes in the real world. Moreover, this investigation reveals that age, heart disease, glucose level, hypertension and marital status are the most significant risk factors. At the same time, the remaining attributes are also essential to obtaining the best performance.

KEYWORDS

Stroke prediction, Machine learning, Classification, Feature selection, Stroke risk factors, Healthcare.

1. INTRODUCTION

Stroke is the second biggest life-threatening disease in the world. It has caused about 11% of deaths worldwide from 2000-2019 [1]-[2]. According to WHO's classifications[1], it is the fourth leading cause of death in low-income countries, the second in lower-middle-income and upper-middle-income countries and the third in high-income countries. In the United States, a stroke happens every 40 seconds, killing one person every (i.e., 1 of every 20 deaths) 3 minutes and 33 seconds [3]. In addition, more than 795,000 people have a stroke, approximately 610,000 of these are the first cases and even a stroke is expected to have a severe long-term disability.

When blood that flows to the brain reduces, there is a lack of nutrients in the cells, quickly leading to cell dysfunction. The symptoms of stroke appear when any part of the brain fails. For example, a core area in a stroke is where blood is almost completely blocked and the cells die within five minutes [4]. There are many reasons for a stroke occurring in a person. These include age, hypertension, diabetes, heart failure, ethnicity, heredity, physical inactivity and peripheral artery disease [5]-[6]. A stroke generally increases with age, but can occur at any period. In 2014, 38% of people hospitalized for a stroke were under 65 and 30% of patients aged 85 and above died from stroke [7].

Stroke is a curable condition that can be considerably reduced in severity if diagnosed or anticipated early. In various investigations and clinical trials, several risk factors for stroke have been found [8]. Proper management and controlled trials, such as preventing high blood pressure, avoiding smoking and

[1] https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death

T. Ahammad is with Department of Computer Science & Engineering, Jagannath University, Bangladesh. ORCID: 0000-0002-0218-1365
Email: tanvir@cse.jnu.ac.bd

alcohol, controlling diabetes, lowering cholesterol, surgery for carotid stenosis and maintaining height and weight adjustment, can reduce the risk of stroke [6], [9]-[11]. Moreover, other diets and mobile technology effectively prevent initial stroke in combination with salt restriction. On the other hand, health agencies can build secondary preventive measures for stroke [12]. Therefore, providing insightful information about stroke prognosis through research from patients' medical history as a tertiary action with personal, medical and secondary management is essential in today's world context.

The inclusion of Artificial Intelligence (AI), especially Machine Learning Algorithm (MLA), has changed the traditional healthcare paradigm into an intelligent health service system. MLAs find hidden patterns from a health data repository and establish models to predict disease for making data-driven healthcare decisions [13]. Predicting the sign of stroke using an MLA is a promising task. Two potential procedures, such as CT scan/MRI and risk factor analysis, can easily monitor the incidence of stroke. Brain imaging can detect real-time stroke on bio-signal data more accurately than risk analysis, as shown in [14]. However, the main drawback of this CT/MRI approach is not anticipating the probability of other diseases (e.g. cardiovascular disease or diabetes). In addition, this approach cannot identify the correlations between the risk factors or the most influential feature importance. Therefore, predictive analysis of risk factors is a prominent approach to observing the likelihood of stroke symptoms.

In recent years, numerous studies have identified predictive analysis of stroke disease using the MLA approach based on the publicly available stroke datasets. In 2019, H. Ahmed et al. [15] examined the presence of stroke with 90% accuracy. Then, P. Govindarajan et al. [16] demonstrated the stroke prognosis for only 507 patients with an accuracy of 96% in 2020. Afterward, in 2021, A. Kumar [17] and T. Tazin et al. [18] showed how to detect stroke using different MLAs with 82% and 95% accuracy, respectively. Finally, in 2022, S. Dev et al. [19] proposed an approach for predictive analysis of stroke risk factors and found four attributes that showed the best accuracy rate, around 80% only. However, the research gap in these studies includes choosing the combination of various pre-processing tasks to improve the quality of classifiers significantly. Moreover, these studies should identify the key risk factors responsible for stroke disease and predict the likelihood of stroke with high-performance MLA models.

This article presents an enhanced approach for identifying possible risk parameters of stroke and predicting its presence in publicly available stroke datasets. First, this approach collects and loads the clinical data containing patients' diagnoses with stroke disease. Next, the dataset is pre-processed and transformed into a standard format to improve the performance of the approach. Then, the best-fit features are identified to find the key risk factors of a stroke. Afterward, ten classification models are used to predict the presence of stroke. Finally, the performance of the classifiers is recorded and compared in terms of accuracy, F1-score, precision, recall and auc_roc to find the best-performing algorithms. The experimental results revealed that Extreme Gradient Boosting (XGB), Gradient Boosting Machine (LGBM), Category Boosting Classifier (CBC) and Adaptive Boosting Classifier (ABC) showed the highest accuracy (97%) of stroke prediction with all feature classifications. In addition, patients' age, cardiovascular disease, diabetes, hypertension and marital status are the most significant risk factors. Overall, the proposed approach demonstrated a higher accuracy of 97% compared to the Machine Learning (ML) models used in existing research [15]-[19] on the same publicly available stroke dataset.

The main contribution of this research is to present an enhanced method that identifies the critical risk factors of stroke and then predicts the possibility of receiving a stroke. In sum, the contributions of this article are as follows:

- Choosing a combination of various pre-processing tasks to improve the quality of classifiers significantly;
- Identifying the best-fit features (risk factors) of the stroke dataset to feed into ML models;
- Ranking key risk factors that are responsible for stroke;
- Achieving the highest percentage of accuracy using advanced gradient boosting-based classifiers that can be the most appropriate ones for physicians to prognose stroke based on the patients' medical history in the real world.

The rest of the paper is structured as follows. First, Section 2 discusses the background of the research. Then, Section 3 represents the description of the dataset, materials and proposed methodology used in this study. Next, Section 4, entitled results and discussion shows the discussion and analysis of the

experimental results. Finally, the paper concludes by suggesting future directions in Section 5, entitled conclusions.

2. RELATED WORKS

Stroke is one of the highest reasons for death globally and causes mental and functioning concerns. So, extensive research is required to find ways to monitor, prevent and treat stroke. The benefits of artificial neural networks (ANNs) and other MLAs have been noticed in the literature to diagnose or predict the occurrence of stroke in a patient [20]. For example, D. Shanthi et al. [21] used an ANN to predict Thromboembolic strokes caused by a thrombus (blood clot) that forms in the arteries delivering blood to the brain. They used stroke data from healthcare datasets with eight attributes of patients. Their investigation improved accuracy to 89%. Although their approach emphasizes prediction accuracy, it is challenging to identify risk factors with a higher performance.

A significant number of research studies have been conducted in the literature to anticipate the possibility of stroke in the human brain using machine-learning (ML) models. First, H. Ahmed et al. [15] used MLAs to identify the presence (90% accuracy) of stroke on the Apache Spark, an open-source distributed processing system used for Big Data workload. Then, G. Sailasya and G. L A. Kumari [17] examined a similar type of study. They compared traditional ML methods and obtained 82% accuracy using the Naive Bayes classifier. Finally, T. Tazin et al. [18] examined how to detect the probability of stroke with a higher accuracy (95%) than in previous studies. However, their methodologies require normalization before feature selection and rank physiological factors to detect strokes more accurately.

Medical imaging and bio-signal analysis are promising research methods to monitor stroke as early as possible. For example, J. Yu et al. [14] developed an AI-based real-time stroke-prediction system on patients' EMG (electromyography, measuring muscle response or activity) bio-signals. They collected and measured real-time left and right biceps femoris (thigh muscle located in the posterior portion or back) and gastrocnemius muscles (large back muscles or back part of the lower leg of humans) from health monitoring devices at 1500 Hz. Their experimental results revealed that the proposed approach could be an alternative to stroke detection with a low-cost diagnosis. However, though their system effectively detects early stroke, it overlooks the risk factors in predicting pre-stroke conditions, because risk-factor analysis shows which parameters are responsible for stroke in advance.

Anticipating the likelihood of a similar type of stroke is a robust approach. This investigation was conducted by L. Amini et al. [22]. They collected 50 different attributes of healthy and sick patients in two hospitals from 2010 to 2011. They used data-mining techniques to classify high-risk groups of patients' history of cardiovascular disease, hyperlipidemia, diabetes, smoking and alcohol consumption. In continuation of predictive stroke analysis, C. Colaka, E. Karaman and M. G. Turtay [23] proposed knowledge discovery from data (KDD) methods on nine attributes. They used 297 data samples (130 sick and 167 healthy persons) and showed the highest accuracy, approximately 93%, by using an ANN. Similarly, L. I. Santos et al. [24] used a decision tree-based ML model to predict the stroke outcomes for the imbalance dataset. They obtained 70% and 78% accuracy to show the significance of their study with the state-of-the-art approach. However, these investigations incorporated small and limited data samples, resulting in poor approximation. In addition, the most significant risk issues of stroke were unrevealed in these studies.

Predictive analysis of risk factors is a promising research approach for stroke disease. S. Dev et al. [19] introduced a method that analyzes and identifies potential physiological attributes related to stroke disease. Using a perceptron neural network, they found four critical risk factors that exhibited the best performance, about 80% accuracy rate. Although they examined the significant risk factors, the accuracy of their approach could improve by choosing a combination of different pre-processing tasks. Besides, they reduced many critical attributes that could give a better predictable rate. D. Paikaray and A. K. Mehta [25] examined a similar approach to predicting stroke before its occurrence. They used nine different ML models in their experiment. They achieved a promising result with an accuracy of 95.10%. Although their experimental result was better than that of S. Dev et al. [19], they could not discover the possible risk factors that may cause a patient's stroke.

Analyzing the effects of risk factors on stroke monitoring is an emerging research trend. For example,

P. Songram and C. Jareanpon [26] showed that people could prevent stroke by predicting its risk factors. They identified seven health issues for a stroke. Using the decision-tree approach, they achieved 74.29% of accuracy in the F1-score. Likewise, R S Jeena and A Sukeshkumar [27] developed a stroke risk-assessment model by detecting relevant predictors. They categorized the risk factors into low-risk, medium-risk and high-risk factors. In addition, Fang et al. [28] used an integrated ML methodology to select the essential features for stroke prognosis. They chose twenty-three parts to predict the acute stroke with an accuracy of 69% only. While these researches have shown the potential of feature selection related to stroke, they have demonstrated a lower accuracy than that obtained in our proposed approach. Moreover, they have identified many attributes that can be difficult to correlate with the probable stroke signs in patients. Therefore, an enhanced approach for identifying ranking-based stroke risk factors and predicting stroke incidence is essential as an alternative to the existing methodologies.

3. MATERIALS AND METHODS

This section represents the description of the stroke dataset, the methodology and the analysis of results from the ten classifiers used in this research.

3.1 Dataset

The dataset used in this research was related to stroke disease. The dataset indicates whether a patient is likely to suffer a stroke based on different parameters, such as gender, age, various diseases and physical conditions. The dataset was publically accessible on the Kaggle2[2] online community platform. It contains 12 different attributes and around 5,110 records or rows of data. Each row comprises relevant patient medical history information, as shown in Table 1. The dataset has 201 missing values in BMI attribute and 1,544 in smoking_status attribute. Moreover, it is a binary classification with a strongly imbalanced dataset involving 4,861class label 0 and 249 class label 1.

Table 1. Summary of stroke dataset.

Attribute Name	Attribute Description
id	Unique identifier of the patient
gender	"Male", "Female" or "Other"
age	Age of the patient
hypertension	0: if the patient doesn't have hypertension; 1: if the patient has hypertension
heart_disease	0: if the patient doesn't have any heart disease; 1: if the patient has a heart disease
ever_married	"No" or "Yes"
work_type	"children", "Gov. job", "Never worked", "Private" or "Self-employed"
Residence_type	"Rural" or "Urban"
avg_glucose_level	Average glucose level (mg/dL) in blood after meal
BMI	Body Mass Index
smoking_status	"formerly smoked", "never smoked", "smokes" or "Unknown". Unknown status indicates that the information is unavailable for this patient
stroke	1: if the patient had a stroke; 0: if the patient had no stroke. (It is the class label attribute)

3.2 Machine-learning Classification Models

This study focuses on identifying risk factors for the binary classification of stroke disease. We employed ten different classification models from various fields of machine learning [29], as shown in Table 2. The models consist of three-tree based methods, including Random Forest (RF) [30], XGB [31] and Decision Tree (DT) [32]; three ensemble boosting approaches, such as LGBM [33], CBC [34] and ABC [35]; one Support Vector Machine (SVM) [36] and neural network-based Multilayer Perceptron (MLP) [37] ; one K-Nearest Neighbor (KNN) [38] and linear statistical-based approach Logistic Regression (LR) [39]. The classifiers are evaluated independently using different performance metrics and the outcomes are recorded for further analysis.

[2] https://www.kaggle.com/fedesoriano/stroke-prediction-dataset

Table 2. Review of different classification models.

Classifiers	Description	Strengths	Weaknesses
RF	It performs random selection of features to build different decision trees and applies voting policy to obtain the final result.	It is efficient for classification problems with numerical and categorical features.	Making predictions is quite slow once they are trained.
XGB	It is an ensemble method that supports various functions, such as classification, regression and ranking.	It is computationally efficient and predicts the result with high accuracy.	It is slow for a large number of classes.
DT	DT is a popular classification approach. It constructs tree data structure, where an internal node denotes the test on an attribute and a leaf node determines the class	It is simple and fast and has good accuracy depending on dataset.	It takes a long time when training the dataset and deals with memory unavailability with respect to large data.
LightGBM	It is a gradient boosting algorithm for classification problems.	It has a faster training speed, higher efficiency and a lower memory utilization. It can also handle large-scale data.	It is prone to overfitting; it can easily overfit small data.
CatBoost	It is a gradient boosting algorithm that predicts with a less amount of time for unseen data.	It is very useful in categorical data without explicit pre- processing.	It needs to construct deep decision trees in order to get better accuracy.
AdaBoost	It is an ensemble boosting classifier by the combination of multiple classifier models to increase accuracy.	It provides high-accuracy outcomes.	It does not perform well with noisy data and outliers.
SVM	It performs classification by setting the hyper-plane that distinguishes between two class labels.	It works very well with a strong margin of segregation for high-dimensional spaces.	It is slow with large datasets.
MLP	It is a feedforward neural network-based classifier, which learns on the non-linear functions for complex data.	It is very powerful and works with high accuracy for both small and large datasets.	The training process is time-consuming to determine the exact parameters for obtaining expected performance.
KNN	It solves classification and regression problems by setting the K-neighbors.	It is a non-parametric algorithm, which implies that certain assumptions must be met in order for it to work.	KNN requires to find tune K-value that may be challenging for large dataset.
LR	It is a statistical model that solves classification and regression problems.	It is easier to extend additional classes and a probabilistic view of class predictions.	The assumption of linearity between the dependent and independent variables is a key constraint of LR.

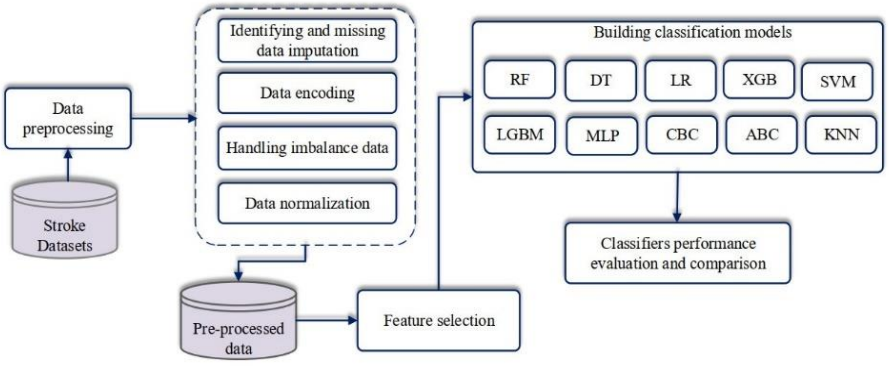


Figure 1. Methodology for the identification of risk factors and prediction of stroke disease.

3.3 Methodology

The proposed approach of this article is a refinement of several methodologies, such as [15]-[18], [40] in the context of stroke-disease analysis. It incorporates six-stage processing phases for identifying and predicting the main risk factors of stroke disease, as illustrated in Figure 1. The stages are stated as follows:

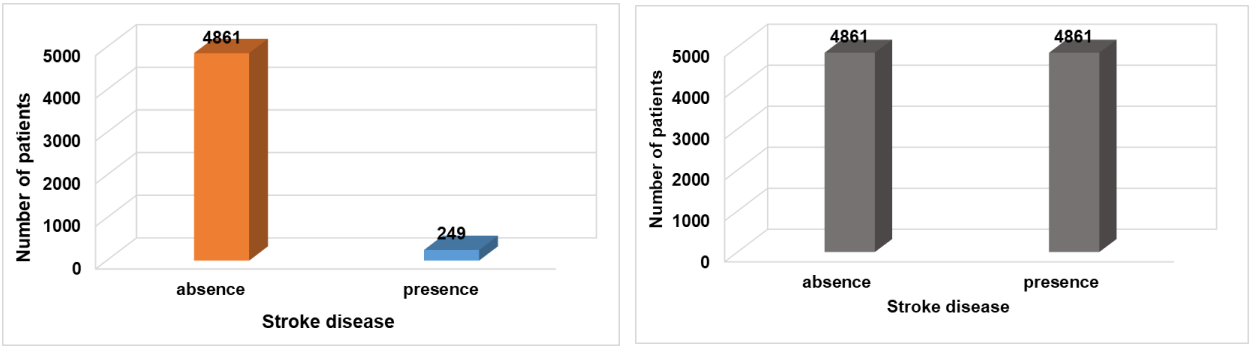
1) **Collecting and loading dataset**. Select and load the target dataset from the health data archive containing patients' medical records related to stroke disease. Since this paper focuses on a publicly accessible stroke dataset, the dataset is first loaded into the program for analysis.

2) **Data pre-processing**. Before feeding target data into the classifiers, this step involves analyzing the datasets to find any inconsistency (e.g. missing values, noise or extreme values). Moreover, this stage transforms the data into a well-formed format to enhance the performance of classifiers. As stated earlier, the stroke dataset has twelve attributes, where *bmi* and *smoking_status* contain missing values. So, these missing values are predicted and replaced by certain values analogous to non-missing data, called missing-data imputation. In continuation of the pre-processing data phases, the columns or attributes containing the categorical or text data are encoded to numeric values so that the ML models can process them properly.

Furthermore, the stroke dataset is rigorously checked to determine the class-label imbalances. As there are a total of 5,110 data records where 249 of them indicate the incidence of a stroke and 4,861 rows indicate the absence of a stroke (Figure 2a), these disparities (imbalance ratio 20:1) may lead many ML models to low predictive accuracy (e.g. metrics like precision and recall) with infrequent class. Consequently, the unbalanced data must be dealt with first to obtain an efficient model. Improved Synthetic Minority Over-sampling Technique (ISMOTE) [41] is possibly a novel approach that selects new samples nearest to the minority-class neighbors; it then balances the minority-class with the majority-class instances. Figure 2 shows the ratio of data samples in the class distribution used in this study. In the final pre-processing stage, the dataset is changed to a standard scale using z-score normalization3[3], as shown in Equation 1.

$$z_{ij} = \frac{f_{ij} - m_i}{sd_i} \tag{1}$$

where, z_{ij}: normalized score j^{th} value of i^{th} feature, f_{ij}: j^{th} value of i^{th} feature, m_i: mean value of i^{th} feature and sd_i: standard deviation of i^{th} feature.



(a) Imbalanced distribution (b) Balanced distribution

Figure 2. Proportion of samples in the number of stroke absence to the number of stroke incidence.

3) **Archiving pre-processed data**. Different preprocessing methods convert the raw data into various understandable formats. For example, various encoding schemes or data-normalization techniques generate distinct data values that may affect the performance of ML models. So, storing all of these formats in a data archive or data files is necessary. In other words, archived data allows ML algorithms to get comprehensible dataset features during the training or learning.

4) **Feature selection**. This stage is essential for deciding the best-fit features for the classifiers' best performance. Firstly, use all the attributes in the target stroke dataset to build and measure the

[3] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.zscore.html

accuracy of the classification models. Next, calculate feature-importance scores using either tree- based classifiers or correlation coefficient; select the top-most n − 1, n − 2, n − 3, ..., 1 features and find classifiers' accuracy, respectively. Finally, a voting procedure is applied to get the best accuracy among all the choices of feature selection. In other words, all combinations of attributes in the dataset are used in the ML models and then recorded as the best-fit feature selection classification models.

5) **Model building**. As stated earlier, ten classification algorithms are used in this study to show the performance of the proposed approach. Therefore, the ratio of data samples used for training and testing purposes is 80:20. Since many ML models have different parameters/variables that control the model's performance, the parameters can not directly predict (e.g. KNN, MLP) from data to obtain the desired accuracy. So, we need to tune the parameters. However, we train all ML models by setting different parameters, grid searching or random searching of model hyperparameters to be learned from data for the best accuracy.

6) **Applying evaluation metrics and performance comparison**. It is the final stage of the proposed methodology. After building the classification models, analyze them using five metrics: accuracy, F1-score, precision, recall and roc_auc (compute area under the receiver operating characteristic curve from prediction scores). Then, the performances of the classifiers are compared based on these criteria. The metrics are defined as follows:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{2}$$

$$Precision = \frac{TP}{TP+FP} \tag{3}$$

$$Sensitivity = Recall = \frac{TP}{TP+FN} \tag{4}$$

$$F1 = \frac{2*Precision*Recall}{Precision+Recall} = \frac{2*TP}{2*TP+FP+FN} \tag{5}$$

$$Specificity = \frac{TN}{FP+TN} \tag{6}$$

$$ROC_{AUC} = Sensitivity - (1 - Specificity) \tag{7}$$

where: $TP = TruePositive$: ML model correctly predicts that a patient has stroke disease.
$TN = TrueNegative$: ML model correctly predicts that a patient has no stroke disease.
$FP = FalsePositive$: ML model incorrectly predicts that a patient has stroke disease.
$FN = FalseNegative$: ML model incorrectly predicts that a patient has no stroke disease.

4. RESULTS AND DISCUSSION

4.1 Exploratory Analysis of Dataset

Exploratory data analysis is necessary to analyze the presence of stroke disease. It is the process of discovering patterns and irregularities and checking premises with the help of summary statistics and visual representations before applying ML models. The stroke dataset used in this study comprises 11 feature attributes and one attribute containing two class labels, shown in Table 1. We did not consider the attribute, *id*, in our analysis, because it does not influence the performance of the classifiers. Since most features are categorical, it is easy to find patterns in the medical history responsible for a patient's stroke.

Figure 3 depicts various distributions of categorical features concerning stroke. For example, Figure 3a illustrates three attributes the value of which is in binary type. Looking closer at this figure, we can see that 13.25% of patients who suffer from hypertension have stroke disease, while the number is below 4% for stroke patients with no hypertension. On the other hand, the number of stroke patients who got married is three times more than those not married. Besides, the ratio of people having a stroke with heart disease is 17% which is more than four times higher than the patients with no heart disease (4.18%). Therefore, heart-disease patients are likely to have a higher risk of stroke than stroke patients with hypertension and those ever married, as shown in Figure 3a. Turning to the attribute work type (Figure 3b), we observe that self-employees (8%) suffer a slightly higher percentage of

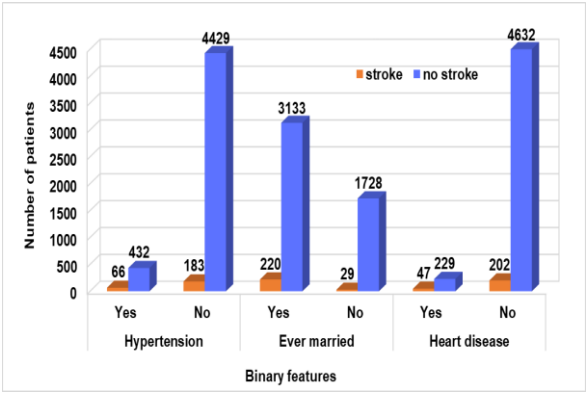
strokes than government (5%) and private (5.09%) workers. Moreover, the stroke patient rate trend seems to be almost similar in residence type and gender groups.

In the stroke dataset, numerical data with missing values can be detected in some entities, as demonstrated in Figure 4. However, to visualize the stroke disease trend, first, numerical features are converted into categorical ones based on predefined rule-based approaches. For instance, the attribute, age, is grouped into four clusters based on reference [42], as stated below:

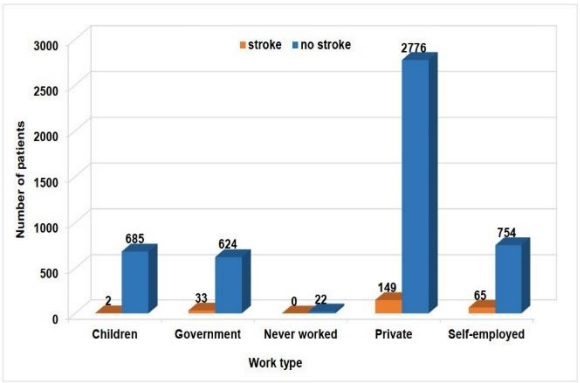
- Child: 0-12 years;
- Adolescent: 13-18 years;
- Adult: 19-59 years;
- Senior Adult: 60 years and above.

Then, the feature bmi containing null/missing values is organized into several categories following the Centers for Disease Control and Prevention (CDC) interpretation44, defined as follows:

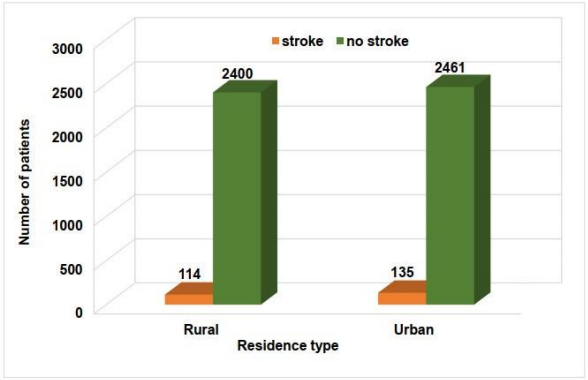
- below 18.5: underweight;
- between 18.5 and 24.9: healthy weight;
- between 25 and 29.9: overweight;
- between 30 and 39.9: obese;
- missing values: null values.



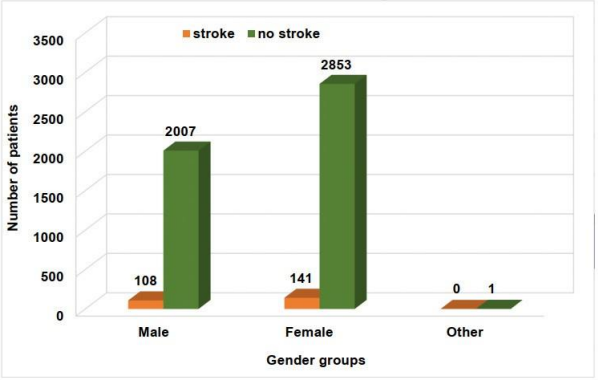
(a) Binary features



(b) Work type status



(c) Residence type



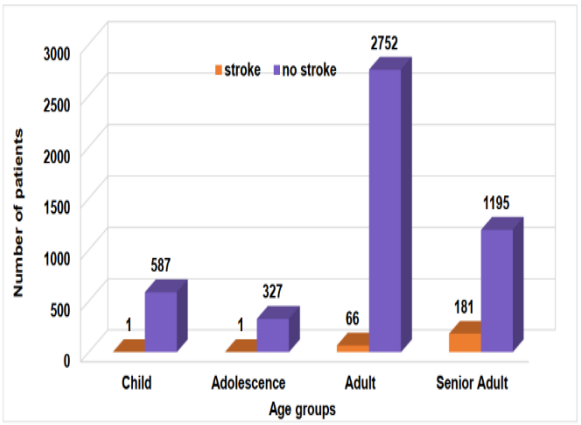
(d) Gender

Figure 3. Representation of categorical attributes.

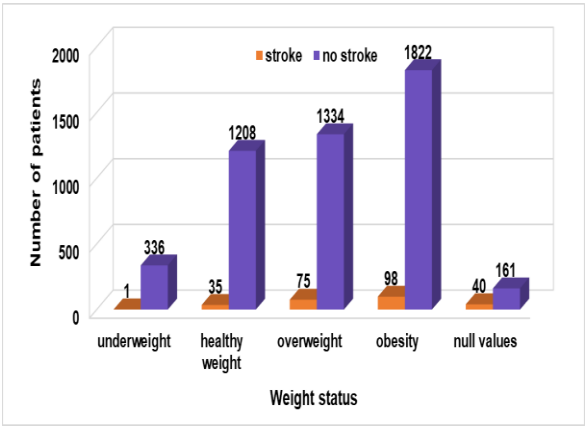
Next, the average glucose level measured after the meal is grouped into three types using CDC report55, as indicated below:

- Diabetes: 200 mg/dL or above;
- Pre-diabetes: 140 to 199 mg/dL;
- Normal: 140 mg/dL or less.

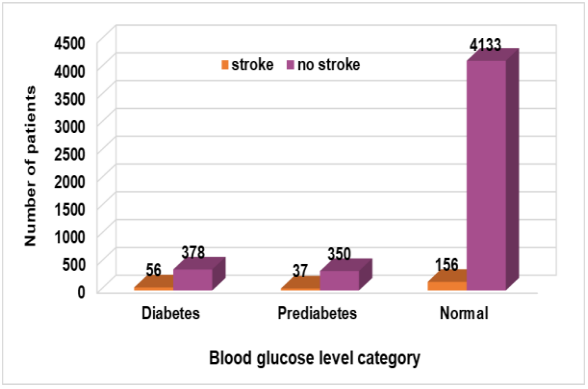
[4] https://www.cdc.gov/healthyweight/assessing/bmi/adult_bmi/
[5] https://www.cdc.gov/diabetes/basics/getting-tested.html



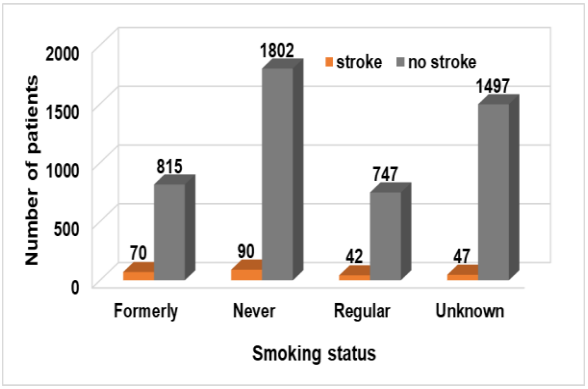
a. Age



b. BMI



c. Average glucose level



d. Smoking status

Figure 4. Representation of numerical features as categorical.

Finally, three categorical values were specified in the original dataset in the smoking status attribute, except for missing values that were later grouped into "Unknown" status, as indicated in Figure 4d. However, if we look at Figure 4, we can notice that senior adults are more likely to suffer from stroke (13%) than other age groups (Figure 4a). Likewise, stroke patients who are overweight and obese have higher numbers than others. In addition, nearly 20% of patients were found to be more likely to suffer from stroke in the missing BMI values, as shown in Figure 4b. Most importantly, the average glucose level in blood is another feature that reveals a noticeable portion, nine and a half percent, of diabetic patients who suffer from stroke (Figure 4c).

Identifying what kind of risk factor can predict a stroke is an important step. In other words, before applying ML models, feature correlation is a practical approach to determine the closeness between features and the target class. This method groups the related health information (e.g. age, bmi or smoking status) to reduce personal data processing, eliminate less essential data and improve the performance of ML models. Figure 5 illustrates the relationship of features with the stroke attribute. It shows that most features are positively correlated with the target variable other than gender and smoking status.

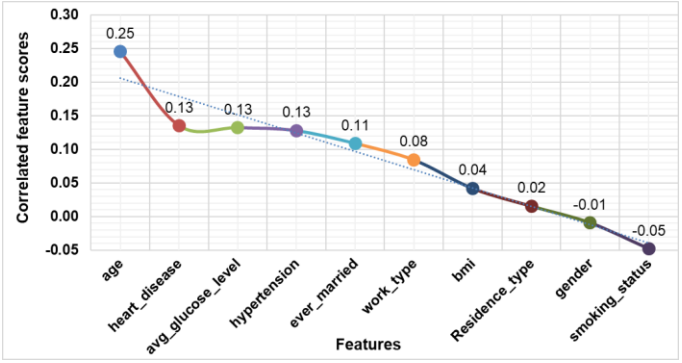
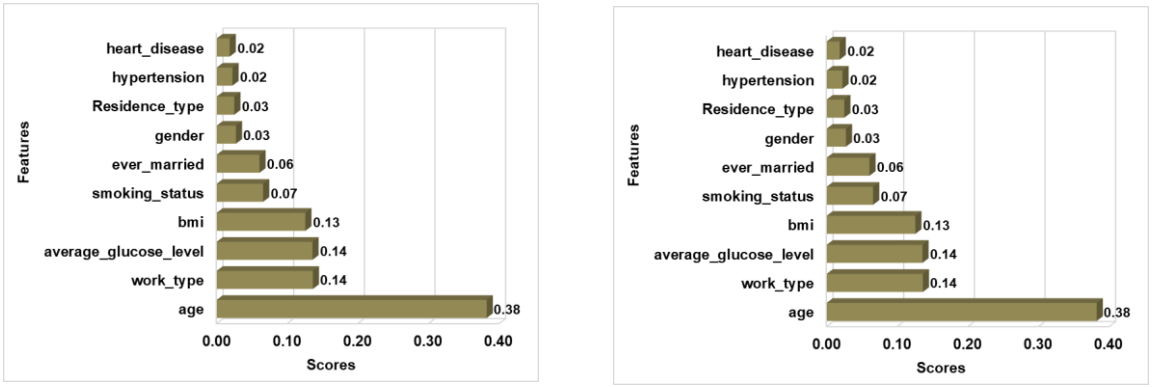


Figure 5. Feature correlation with the *stroke* attribute.

4.2 Performance Evaluation of Classifiers

The performance of the ML classification models is measured using all features and top-most main attributes in the dataset, as mentioned in the section containing the research methodology. The top-most important attributes are selected and ranked according to feature-importance scores by RF and XGB classifiers (Figure 6). A higher score implies that the specific feature will significantly impact the classification model. However, if we observe closely Figure 6a and Figure 6b, we can see that the two approaches, RF and XGB, generated different feature rankings despite a similar tree-based paradigm. In addition, three are the most common features in the top five scores. So, we applied these top-most features in classification. We also considered listing key attributes from correlated feature scores.

As said previously, first, we tested and evaluated the performance of ten classifiers on all features from the stroke dataset. Table 3 shows the results from the analysis in the experiment. We can see that gradient boosting-based classifiers, including XGB, LGBM, CBC and ABC, showed the best accuracy (97%). On the other hand, the LR model gave the lowest result in terms of accuracy (80%), F1-score (86%), precision (81%), recall (86%) and roc_auc (86%), respectively. Besides, RF and MLP showed the second-highest accuracy, whereas SVM and K-NN performed similarly. However, Figure 7 depicts the performance analysis of all classifiers used in this study on the stroke dataset. It reveals that the performance rate of most classifiers slightly fluctuates between 94% and 97% except for the LR model. Overall, in all feature selections, gradient and ensemble boosting-tree-based ML models exhibit higher performance (97%) for stroke-disease detection compared to the ML models used in the existing research studies [15]-[19].



a. RF for feature importance

b. XGB for feature importance

Figure 6. Ranking of features for stroke-disease prediction.

Table 3. Train-test performance evaluation of classifiers on all feature sets.

	Performance-evaluation metrics				
Classifier	**Accuracy (%)**	**F1-score (%)**	**Precision (%)**	**Recall (%)**	**ROC_AUC (%)**
RF	96.4	96	96	96	96
XGB	**97**	**97**	**97**	**97**	**97**
SVM	95	95	95	95	95
DT	94	94	94	94	94
LGBM	**97**	**97**	**97**	**97**	**97**
CBC	**97**	**97**	**97**	**97**	**97**
ABC	**97**	**97**	**97**	**97**	**97**
MLP	96	96	96	96	96
K-NN	95	95	95	95	95
LR	80	81	81	86	86

Top-most attribute selection is another benchmark for predicting stroke disease. So, we selected different subsets (except null and all feature sets) of attributes on the target dataset. Based on the order indicated in Figure 5, we found the seven most essential feature classification presented the best accuracy. Table 4 summarizes the obtained results on the performance of ten classifiers. It shows that XGB, LGBM and CBC have achieved the highest performance, similar to the results in [18] but better than the results in works [15]-[17]. Figure 8 illustrates the visual representation of stroke prediction accomplishment on the top-most seven-feature dataset. We can see that the performance rate starts with a rising trend from RF to XGB.

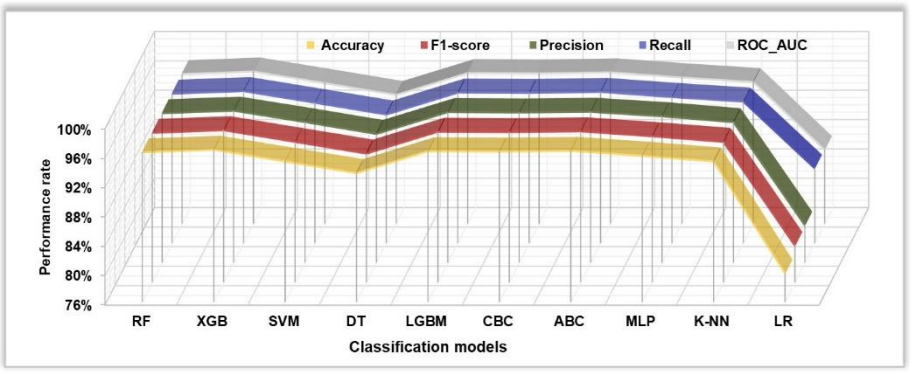


Figure 7. Train-test performance analysis on stroke dataset for all features.

Table 4. Train-test performance evaluation of classifiers on top-most feature sets.

	Performance evaluation metrics				
Classifier	**Accuracy (%)**	**F1-score (%)**	**Precision (%)**	**Recall (%)**	**ROC_AUC (%)**
RF	94	94	94	94	94
XGB	**96**	**96**	**96**	**96**	**96**
SVM	91	91	91	91	91
DT	92	92	92	92	92
LGBM	**96**	**96**	**96**	**96**	**96**
CBC	**96**	**96**	**96**	**96**	**96**
ABC	94	94	94	94	94
MLP	92	92	93	92	92
K-NN	92	92	92	92	92
LR	78	79	81	83	85

Then, it falls and remains constant for SVM and DT; it increases again sharply and reaches the highest peak at 96% for LGBM and CBC; afterward, it presents a downward trend and reaches the lowest point (78%). Therefore, LR was the lowest-performing model, whereas XGB, LGBM and CBC were the best-performing models on the top-most seven features.

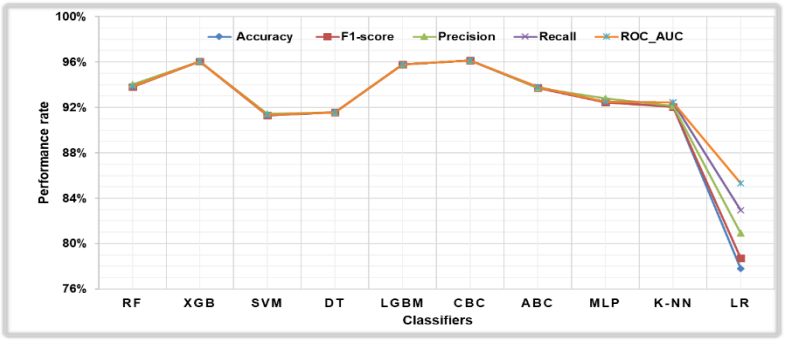


Figure 8. Performance from the top-most seven features.

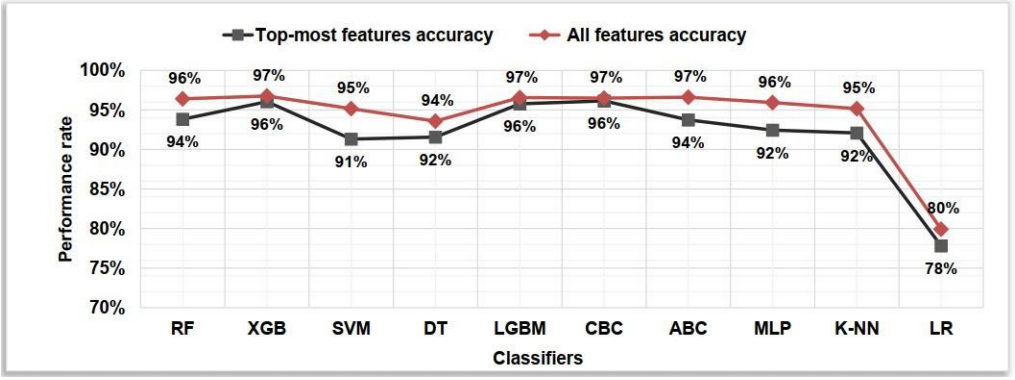


Figure 9. Comparison of classifiers' performance on different feature selections.

The implication of this research lies in finding the best performance from the ten ML classifiers to determine the subset of attributes for stroke-disease detection and prevention. We identified two test cases, including all features and the top-most seven attributes. We also separately compared the results obtained from ML classifiers on these two types of attribute selection. As a result, the attribute/feature selections show different patterns, as depicted in Figure 9. Looking at the graph, we can observe that the results obtained from the ten ML models demonstrate an average of 2.1% better accuracy for all features than the top-most seven features. In addition, SVM, ABC, MLP and K-NN models revealed the most asymmetry differences (3%-4% accuracy deviations) in these classifications. Therefore, considering our analysis, we conclude that the ML classification models performed sufficiently on all attributes of the stroke dataset. In other words, the classifiers do not perform considerably well by selecting different subsets of attributes rather than all (ten features in the stroke dataset).

As stated above, this article presents an enhanced method that performs well for all feature classifications. So, we compare the best-performing results of this article with those of other approaches. Table 5 represents a summary of the classification accuracy of different methods in several studies, including this article. The table shows that RF and DT classifiers were common in all papers for stroke prediction. We can also see from the table that every classification model used in this article exhibits a better accuracy rate than others, despite the similarity in [18] for the DT classifier. One significant point is that none of the existing research studies used gradient and ensemble boosting-tree-based (except [16]) classifiers. In other words, gradient and ensemble boosting-tree-based ML models showed the highest percentage (97%) of stroke prognosis on the same stroke dataset used in previous studies. In conclusion, this article outperformed previous works [15]-[19] using the methodology of six-stage processing phases.

Table 5. Performance comparison of classifiers in different studies.

ML models	Ref. [15]	Ref. [16]	Ref. [17]	Ref. [18]	Ref. [19]	This article
RF	90%	90.9%	72%	96%	75%	96.4%
XGB						**97%**
SVM	77%	91.5%	78.6%		68%	95%
DT	79%	90.7%	77.5%	94%	74%	94%
LGBM						**97%**
CBC						**97%**
ABC		91.5%				**97%**
MLP		95%			80%	96%
K-NN			77.4%			95%
LR	77%	90.6%	77.5%	79%		80%

We found in our analysis that all the medical records presented in the stroke dataset are essential for the stroke disease of the patients. We also ranked the patients' health history in terms of feature importance. Age, heart disease, diabetes, high blood pressure and marital status are considered the most critical factors for a patient's stroke. Besides, the type of workplace and the ratio of height and people's weight are also significant factors. Although patients' residential environment, gender type or smoking habits demonstrated less importance in the analysis, we can not ignore them to detect and prevent stroke.

5. CONCLUSIONS

Stroke is one of the deadly diseases at the global level. Healthcare providers should identify its causes and take preventive measures as early as possible to avoid complications. However, it is a critically challenging problem for healthcare professionals and researchers. This research focuses on an enhanced approach for identifying the risk factors and detecting the presence of stroke for clinical stroke datasets using ML models to solve the issues. First, we analyzed the dataset to find any inconsistencies and discover hidden patterns. Next, we selected different subsets of features to identify and rank stroke risk factors for classification. Then, we relied on ten ML classification models to predict the presence of stroke using the train-test splitting technique. Finally, we evaluated the performance of classifiers using five metrics, including accuracy, precision, F1-score, recall and roc_auc.

We compared the results of the ten ML models in all features and the top-most seven feature

classifications. We observed that the classifiers performed differently (2.1% divergence) on these two feature selections. XGB, LGBM, CBC and ABC models showed the highest accuracy rates in all feature (ten attributes) classification, whereas XGB, LGBM and CBC were the most accurate ones for the top-most seven attributes. We also showed that every classification model used in this article exhibits a higher accuracy rate than other studies in most of the cases. Overall, we obtained a higher accuracy of 97% than the existing approaches on the stroke dataset using gradient and ensemble boosting-tree-based classifiers. Therefore, healthcare providers can use these classifiers that are the most suited for predicting stroke based on the medical history of a patient in the real world.

Furthermore, our experimental results revealed that age, heart disease, glucose level, hypertension and marital status are significant risk factors. Other attributes, such as employment variety, bmi, residential status, gender and smoking status, are essential in predicting stroke to achieve the best accuracy. However, in the future, an intelligent stroke -diagnosis and- monitoring system will be proposed to capture real-time health status (e.g. blood pressure, pulse rate/ECG and glucose level) and then predict the probability of stroke. Moreover, the system will apply to analyzing other diseases (e.g. heart disease and kidney disease).

ACKNOWLEDGMENTS

I want to express my gratitude to the Department of Computer Science and Engineering, Jagannath University, Dhaka-100, Bangladesh, for allowing me to use the lab to conduct the work.

CONFLICTS OF INTEREST

There are no conflicts of interest regarding the publication of this paper.

REFERENCES

[1] C. O. Johnson, M. Nguyen, G. A. Roth et al., "Global, Regional and National Burden of Stroke, 1990–2016: A Systematic Analysis for the Global Burden of Disease Study 2016," The Lancet Neurology, vol. 18, no. 5, pp. 439–458, 2019.

[2] B. C. Campbell, D. A. De Silva, M. R. Macleod, S. B. Coutts, L. H. Schwamm, S. M. Davis and G. A. Donnan, "Ischaemic Stroke," Nature Reviews Disease Primers, vol. 5, no. 1, pp. 1–22, 2019.

[3] S. S. Virani, A. Alonso, H. J. Aparicio et al., "Heart Disease and Stroke Statistics—2021 Update: A Report from the American Heart Association," Circulation, vol. 143, no. 8, pp. e254–e743, 2021.

[4] A. Subudhi, M. Dash and S. Sabut, "Automated Segmentation and Classification of Brain Stroke Using Expectation-maximization and Random Forest Classifier," Biocybernetics and Biomedical Engineering, vol. 40, no. 1, pp. 277–289, 2020.

[5] J. J. Noubiap, V. F. Feteh, M. E. Middeldorp, J. L. Fitzgerald, G. Thomas, T. Kleinig, D. H. Lau and P. Sanders, "A Meta-analysis of Clinical Risk Factors for Stroke in Anticoagulant-Naïve Patients with Atrial Fibrillation," EP Europace, vol. 23, no. 10, pp. 1528–1538, 2021.

[6] M. S. Elkind and R. L. Sacco, "Stroke Risk Factors and Stroke Prevention," Seminars in Neurology, vol. 18, no. 04, pp. 429–440, Thieme Medical Publishers, Inc., 1998,.

[7] G. Jackson and K. Chari, "National Hospital Care Survey Demonstration Projects: Stroke Inpatient Hospitalizations," Natl Health Stat Report, vol. 132, pp. 1-11, National Library of Medicine, 2019.

[8] V. Malik, A. N. Ganesan, J. B. Selvanayagam, D. P. Chew and A. D. McGavigan, "Is Atrial Fibrillation a Stroke Risk Factor or Risk Marker? An Appraisal Using the Bradford Hill Framework for Causality," Heart, Lung and Circulation, vol. 29, no. 1, pp. 86–93, 2020.

[9] H.-J. Lin, J.-H. Yeh, M.-T. Hsieh and C.-Y. Hsu, "Continuous Positive Airway Pressure with Good Adherence Can Reduce Risk of Stroke in Patients with Moderate to Severe Obstructive Sleep Apnea: An Updated Systematic Review and Meta-analysis," Sleep Medicine Reviews, vol. 54, p. 101354, 2020.

[10] K. Furie, "Epidemiology and Primary Prevention of Stroke," CONTINUUM: Lifelong Learning in Neurology, vol. 26, no. 2, pp. 260–267, 2020.

[11] C. English, L. MacDonald-Wicks, A. Patterson, J. Attia and G. J. Hankey, "The Role of Diet in Secondary Stroke Prevention," The Lancet Neurology, vol. 20, no. 2, pp. 150–160, 2021.

[12] J. D. Pandian, S. L. Gall, M. P. Kate et al., "Prevention of Stroke: A Global Perspective," The Lancet, vol. 392, no. 10154, pp. 1269–1278, 2018.

[13] K. Shailaja, B. Seetharamulu and M. Jabbar, "Machine Learning in Healthcare: A Review," Proc. of the 2nd IEEE International Conference on Electronics, Communication and Aerospace Technology (ICECA), pp. 910–914, Coimbatore, India, 2018.

[14] J. Yu, S. Park, S.-H. Kwon, C. M. B. Ho, C.-S. Pyo and H. Lee, "Ai-based Stroke Disease Prediction

System Using Real-time Electromyography Signals," Applied Sciences, vol. 10, no. 19, p. 6791, 2020.

[15] A. A. Ali, "Stroke Prediction Using Distributed Machine Learning Based on Apache Spark," Stroke, vol. 28, no. 15, pp. 89–97, 2019.

[16] P. Govindarajan, R. K. Soundarapandian, A. H. Gandomi et al., "Classification of Stroke Disease Using Machine Learning Algorithms," Neural Computing and Applications, vol. 32, no. 3, pp. 817–828, 2020.

[17] G. Sailasya and G. L. A. Kumari, "Analyzing the Performance of Stroke Prediction Using ML Classification Algorithms," International Journal of Advanced Computer Science and Applications, vol. 12, no. 6, pp. 539–545, 2021.

[18] T. Tazin, M. N. Alam, N. N. Dola, M. S. Bari, S. Bourouis and M. Monirujjaman Khan, "Stroke Disease Detection and Prediction Using Robust Learning Approaches," Journal of Healthcare Engineering, vol. 2021, Article ID 7633381, 2021.

[19] S. Dev, H. Wang, C. S. Nwosu, N. Jain, B. Veeravalli and D. John, "A Predictive Analytics Approach for Stroke Prediction Using Machine Learning and Neural Networks," Healthcare Analytics, vol. 2, p. 100032, 2022.

[20] N. Kasabov, V. Feigin, Z.-G. Hou, Y. Chen, L. Liang, R. Krishnamurthi, M. Othman and P. Parmar, "Evolving Spiking Neural Networks for Personalized Modeling, Classification and Prediction of Spatio-temporal Patterns with a Case Study on Stroke," Neurocomputing, vol. 134, pp. 269–279, 2014.

[21] D. Shanthi, G. Sahoo and N. Saravanan, "Designing an Artificial Neural Network Model for the Prediction of Thrombo-embolic Stroke," International Journal of Biometrics and Bioinformatics (IJBB), vol. 3, no. 1, pp. 10–18, 2009.

[22] L. Amini, R. Azarpazhouh, M. T. Farzadfar, S. A. Mousavi, F. Jazaieri, F. Khorvash, R. Norouzi and N. Toghianfar, "Prediction and Control of Stroke by Data Mining," International Journal of Preventive Medicine, vol. 4, no. Suppl. 2, p. S245, 2013.

[23] C. Colak, E. Karaman and M. G. Turtay, "Application of Knowledge Discovery Process on the Prediction of Stroke," Computer Methods and Programs in Biomedicine, vol. 119, no. 3, pp. 181–185, 2015.

[24] L. I. Santos, M. O. Camargos, M. F. S. V. D'Angelo et al., "Decision Tree and Artificial Immune Systems for Stroke Prediction in Imbalanced Data," Expert Systems with Applications, vol. 191, p. 116221, 2022.

[25] D. Paikaray and A. K. Mehta, "An Extensive Approach towards Heart Stroke Prediction Using Machine Learning with Ensemble Classifier," Proc. of the International Conference on Paradigms of Communication, Computing and Data Sciences, pp. 767–777, Springer, 2022.

[26] P. Songram and C. Jareanpon, "A Study of Features Affecting on Stroke Prediction Using Machine Learning," Proc. of the International Conference on Multi-disciplinary Trends in Artificial Intelligence, pp. 216–225, Springer, 2019.

[27] R. S. Jeena and A. Sukeshkumar, "Development of a Stroke Risk Assessment Model for a Small Population in South Kerala Using Logistic Regression," Proc. of TENCON 2019-2019 IEEE Region 10 Conference (TENCON), pp. 350–355, Kochi, India, 2019.

[28] G. Fang, W. Liu and L. Wang, "A Machine Learning Approach to Select Features Important to Stroke Prognosis," Computational Biology and Chemistry, vol. 88, p. 107316, 2020.

[29] L. R. Guarneros-Nolasco, N. A. Cruz-Ramos, G. Alor-Hernández, L. Rodríguez-Mazahua and J. L. Sánchez-Cervantes, "Identifying the Main Risk Factors for Cardiovascular Diseases Prediction Using Machine Learning Algorithms," Mathematics, vol. 9, no. 20, p. 2537, 2021.

[30] A. Parmar, R. Katariya and V. Patel, "A Review on Random Forest: An Ensemble Classifier," Proc. of the International Conference on Intelligent Data Communication Technologies and Internet of Things, pp. 758–763, Springer, 2018.

[31] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho et al., "Xgboost: Extreme Gradient Boosting," R Package Version 0.4-2, vol. 1, no. 4, pp. 1–4, 2015.

[32] M. Salman Saeed, M. W. Mustafa, U. U. Sheikh, T. A. Jumani, I. Khan, S. Atawneh and N. N. Hamadneh, "An Efficient Boosted C5. 0 Decision-tree-based Classification Approach for Detecting Nontechnical Losses in Power Utilities," Energies, vol. 13, no. 12, p. 3242, 2020.

[33] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye and T.-Y. Liu, "Lightgbm: A Highly Efficient Gradient Boosting Decision Tree," Advances in Neural Information Processing Systems, vol. 30, pp. 3146–3154, 2017.

[34] J. T. Hancock and T. M. Khoshgoftaar, "Catboost for Big Data: An Interdisciplinary Review," Journal of Big Data, vol. 7, no. 1, pp. 1–45, 2020.

[35] W. Wang and D. Sun, "The Improved Adaboost Algorithms for Imbalanced Data Classification," Information Sciences, vol. 563, pp. 358–374, 2021.

[36] S. Suthaharan, "Support Vector Machine," Proc. of Machine Learning Models and Algorithms for Big Data Classification, pp. 207–235, Springer, 2016.

[37] S. Wan, Y. Liang, Y. Zhang and M. Guizani, "Deep Multi-layer Perceptron Classifier for Behavior Analysis to Estimate Parkinson's Disease Severity Using Smartphones," IEEE Access, vol. 6, pp. 36 825–36 833, 2018.

[38] H. Saadatfar, S. Khosravi, J. H. Joloudari, A. Mosavi and S. Shamshirband, "A New K-nearest Neighbors

Classifier for Big Data Based on Efficient Data Pruning," Mathematics, vol. 8, no. 2, p. 286, 2020.
[39] K. Shah, H. Patel, D. Sanghvi and M. Shah, "A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification," Augmented Human Research, vol. 5, no. 1, pp. 1–16, 2020.
[40] J. Yu, S. Park, S.-H. Kwon, C. M. B. Ho, C.-S. Pyo and H. Lee, "Ai-based Stroke Disease Prediction System Using Real-time Electromyography Signals," Applied Sciences, vol. 10, no. 19, p. 6791, 2020.
[41] S. Wang, Y. Dai, J. Shen and J. Xuan, "Research on Expansion and Classification of Imbalanced Data Based on Smote Algorithm," Scientific Reports, vol. 11, no. 1, pp. 1–11, 2021.
[42] J. Nithyashri and G. Kulanthaivel, "Classification of Human Age Based on Neural Network Using FG-net Aging Database and Wavelets," Proc. of the 4th IEEE International Conference on Advanced Computing (ICoAC), pp. 1–5, Chennai, India, 2012.

ملخص البحث:

النّوبة القلبية حالة تهدّد الحياة، وتعدّ ثاني أسباب الوفاة عالمياً. وهي مشكلة تنطوي على تحدٍّ في ميدان الصّحة العامة في القرن 21 للعاملين والباحثين في حقل الرعاية الصحية. لذا فإنّ الرّصد الملائم للنّوبة القلبية يمكن أن يقود الى منعها أو تخفيف شدّتها. هناك العديد من الطّرق التي ركزت على توقّع النّوبات القلبية لدا المرضى. والكثير من تلك الطّرق حقّقت معدّل دقّةٍ عالياً، وصل الى ما يقرب من 90% على مجموعات البيانات المتاحة للعموم. ويمكن لجمْع مهامّ تدريبٍ قبْلي متنوعة أن يحسّن على نحوٍ ملموس جودة المصنِّفات التي تشكل مجالاً يحتاج الى البحث. مع تحديد الباحثين عوامل الخطورة الرئيسية للنّوبة القلبية واستخدام مصنِّفات متقدّمة لتوقّع احتمال الإصابة بالنّوبة القلبية.

تقدم هذه الورقة طريقة محسَّنة لتحديد عوامل الخطورة المحتملة وتوقّع الإصابة بالنّوبة القلبية، وذلك باستخدام إحدى مجموعات البيانات المتاحة للعموم. وتسعى الطّريقة المقترحة الى ردم فجواتٍ في الأدبيّات السّابقة المتعلّقة بالموضوع. وتستخدم الدراسة الحالية عشرة نماذج تصنيف تشمل مصنِّفات معزّزة متقدّمة للكشف عن الإصابة بالنّوبة القلبية. وقد تمّ تحليل أداء المصنِّفات على جميع مجموعات البيانات الفرعية الممكنة المتعلّقة باختيارات الخصائص/السِّمات، بأخذ خمسة مقاييس بعين الاعتبار، لتحديد الخوارزميات الأفضل أداءً. وبينت النّتائج التجريبية أنّ الطريقة المقترحة حقّقت الدّقة الأعلى على تصنيفات السِّمات كافّةً. وكان الإنجاز الحقيقي لهذه الدراسة تحقيق نسبة دقّة أعلى (97% باستخدام مصنِّفات معزّزة) مقارنة بالطّرق الأخرى. وعليه يمكن للأطباء الاستفادة من الطريقة المقترحة في توقّع الإصابة النّوبة القلبية في العالم الحقيقي. وأوضحت النتائج أنّ أبرز عوامل الخطورة المرتبطة بالنّوبة القلبية هي: العمر، وأمراض القلب، ومستوى الغلوكوز، وإرتفاع ضغط الدّم، والحالة الاجتماعية. وفي الوقت ذاته، فإنّ السِّمات الأخرى أساسية للحصول على الأداء الأفضل.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 08, No. 03, September 2022.

RAT SWARM OPTIMIZER FOR DATA CLUSTERING

Ibrahim Zebiri, Djamel Zeghida and Mohammed Redjimi

(Received: 1-Jun.-2022, Revised: 27-Jul.-2022, Accepted: 18-Aug.-2022)

ABSTRACT

Rat Swarm Optimizer (RSO) is one of the newest swarm intelligence optimization algorithms that is inspired from the behaviors of chasing and fighting of rats in nature. In this paper, we will apply the RSO to one of the most challenging problems, which is data clustering. The search capability of RSO is used here to find the best cluster centers. The proposed RSO algorithm for clustering (RSOC) is tested on several benchmarks and compared to some other optimization algorithms for data clustering, including some well- known and powerful algorithms such as Particle Swarm Optimization (PSO) and Genetic Algorithm (GA), as well as other recent algorithms, such as the Hybridization of Krill Herd Algorithm and harmony search (H-KHA), hybrid Harris Hawks Optimization with differential evolution (H-HHO) and Multi-Verse Optimizer (MVO). Results are validated through a bunch of measures: homogeneity, completeness, v-measure, purity and error rate. The computational results are encouraging, where they demonstrate the effectiveness of RSOC over other clustering techniques.

KEYWORDS

Rat swarm optimization (RSO), Swarm intelligence, Cluster analysis, Clustering.

1. INTRODUCTION

Data clustering is an important procedure in data mining [1]-[3]. It consists of dividing a given set of unlabeled data (objects) into finite groups of similar objects. Data clustering has been widely used in several fields such as: image processing, pattern recognition, intrusion detection, biology, medical fields, among others [1]-[6]. There are many categorizations of data-clustering techniques depending on some criteria [2], [7]-[8], such as categorizing data clustering into hard (crisp) and fuzzy clustering. In hard clustering, an object cannot be a part of more than one cluster. However, in fuzzy clustering, an object can be a part of multiple clusters with certain values that indicate the degree of membership to each cluster [2], [9]. Another well-known categorization is partitional and hierarchical clustering [2], [7]-[8]. Hierarchical clustering clusters data progressively making a clusters hierarchy, generally with each object as a cluster at the bottom stage and the whole dataset as a cluster at the top stage; between these two stages, there is a bunch of other stages, where in each stage there is a different number of clusters. Each stage can be used as the final clustering, where the choice of the final clustering (stage) may depend on the number of clusters or any other criterion, such as the distance between clusters. Partitional clustering, however, divides the dataset directly into a certain number of clusters [1]-[3], [8]. In this work, we are interested in partitional clustering. The most known technique of this type is k-means [1]-[2], 10].

Data clustering is considered as an optimization problem [2], as it is impossible in most cases to find the global optimal solution with exact methods. For a machine that can verify a million solutions per second, to test all possibilities of clustering a dataset of 50 objects in three clusters, it would take more than 3 billion years. Thus, the need of powerful (efficient and effective) methods that can find a good solution near to the best one in acceptable time is indispensable. Nature-inspired metaheuristics are optimal tools for such problems [2]. Mainly, they can be categorized into four general types: evolutionary-based algorithms, swarm intelligence-based algorithms, human-based algorithms and physical and chemical-based algorithms [11]. Swarm intelligence-based algorithms are methods inspired from the intelligence shown by swarms in nature. They mimic their collective intelligent behavior of finding food, fighting, defending, hunting, …etc. to explore and find solutions to optimization problems. Ant Colony Optimization (ACO) [12] and Particle Swarm Optimization (PSO) [13] are examples of swarm-intelligence techniques.

Metaheuristics has been widely applied to the clustering problem. Selim and Al-Sultan [14] applied simulated annealing (SA) to clustering. Al-Sultan [15] proposed a tabu-search (TS) [16]-[17] approach for data clustering. It was compared to SA and k-means and it outperformed them on almost all datasets.

I. Zebiri, D. Zeghida and M. Redjimi are with Department of Computer Science, Université 20 Août 1955, Skikda, Algeria. Emails: i.zebiri@univ-skikda.dz,dj.zeghida@gmail.com and m.redjimi@univ-skikda.dz

Genetic algorithm (GA) [18]-[19] is widely applied to this problem [2], [20]-[22]. Shelokar et al. [23] developed an ant-colony approach, where it was compared to SA, TS and GA and the results showed the power of the mechanisms of this approach. In [24], Jinchao et al. proposed a novel artificial bee colony (ABC) [25] based on k-modes (ABC-K-modes) for clustering of categorical data. The proposed algorithm was tested on several datasets and compared to some other popular algorithms for categorical data, where ABC-K-modes outperformed the algorithms compared with in all but few datasets. In [26]-[28], some applications of PSO to data clustering are demonstrated. In [29], authors proposed a hybrid PSO and grey wolf optimizer (GWO) [30] to take advantage of both mechanisms of PSO and GWO and applied it to data clustering. Kumar et al. [31] developed a grey wolf algorithm-based clustering (GWAC) technique, where GWO was applied to find the optimal center for each cluster and k-means to cluster data. The proposed algorithm was tested on both artificial and real datasets and compared with other algorithms. In [32], a magnetic optimization algorithm for data clustering (MOAC) was proposed. The algorithm was tested on eleven datasets and compared to five algorithms, where MOAC showed better results than other algorithms in general. The authors in [33] proposed an enhanced version of black hole algorithm (LBH) and applied it to data clustering. The proposed algorithm was tested on six real datasets and com-pared with nine other algorithms, where it outperformed them in all datasets. In [34], authors hybridized GWO with TS (GWOTS). TS was used to search for optimal solutions near the best ones. GWOTS was tested on several datasets and compared to other algorithms including, GWO and TS, where the results showed the effectiveness of the hybrid method. Aljarah et al. [35] applied multi-verse optimizer (MVO) [36] to data clustering and tested it on several datasets with four measures. MVO outperformed the other algorithms compared with in almost all datasets.

Rat Swarm Optimizer (RSO) [37] is a novel swarm intelligence-based algorithm, which mimics the behavior of rats in chasing and fighting prey in nature. It was applied to several optimization problems [38]-[42]. In this paper, we will apply this method to the clustering problem. The performance of Rat Swarm Optimizer for Clustering (RSOC) has been tested on several various real benchmarks to show its performance.

The remainder of this work is structured as follows: Section 2 introduces the data-clustering problem briefly. Section 3 describes the RSO. The adaptation of the proposed RSO to the clustering problem is presented in Section 4. Finally, experimental results and their discussion are provided in Section 5. Section 6 concludes the paper and opens some horizons for future research.

2. DATA CLUSTERING

Data Clustering is the task of grouping a set of unlabeled data D in k groups called clusters C = (C1, C2,..., Ck), based on some distance or similarity measurements, such as Euclidean and Manhattan distances. Each object should be a member of one and only one cluster and a cluster should at least have a member [2], [4], [10], [43]:

$$\forall i, j \in \{1, \dots, k\} \; and \; i \neq j, C_i \cap C_j = 0$$
$$U_{i=1}^{k} C_i = D$$
$$\forall i \in \{1, \dots, k\}, C_i \neq 0$$

Objects of the same cluster should be closer to each other or similar, while objects from different clusters should be dissimilar or distant. Thus, the problem of clustering can be reformulated as: minimizing the intracluster distances and maximizing the intercluster distances. The Euclidean distance between two objects *x* and *y* is defined as follows:

$$d_{Euc}(x, y) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2} \quad (1)$$

where, x_i and y_i are respectively the i^{th} attributes of *x* and *y*.

Clustering techniques need to be evaluated to reveal their efficacy. Algorithm efficacy is generally measured by two main measures: performance and effectiveness [2]. Performance measures are generally used to compare the efficiency (computational time) of algorithms, without caring about the quality of results. Algorithms to be compared should be applied on the same programming language, tested on the same benchmark and executed on the same machine. On the other hand, effectiveness measures are used to assess the quality of results. Generally, there are three main types of effectiveness measures: internal, external and relative measures [2], [44]-[46]. Internal indices (intrinsic indices)

measure the validity using the information intrinsic to data. Sum of intracluster distances is an example of this type. However, external indices (extrinsic indices) measure the validity of the clustering results using some external information (ground truth), such as the class distribution of the clustered dataset [1]-[2], [47]-[48]. Homogeneity, completeness, v-measure, purity and error rate are external indices, which are, respectively, defined as follows:

$$Homogeneity = 1 - \frac{H(C|L)}{H(C)} \tag{2}$$

$$Completness = 1 - \frac{H(L|C)}{H(L)} \tag{3}$$

where,

$$H(C|L) = -\sum_{i=1}^{k}\sum_{j=1}^{q}\frac{n_{ij}}{n}.\log(\frac{n_{ij}}{n_j})$$

$$H(C) = -\sum_{i=1}^{k}\frac{n_i}{n}.\log\left(\frac{n_i}{n}\right)$$

$$V - measure = 2.\frac{Homogeneity.Completness}{Homogeneity+Completness} \tag{4}$$

$$Purity = \frac{1}{n}\sum_{i=1}^{k} max_j(n_{ij}) \tag{5}$$

$$ErrorRate = \frac{Number\ of\ misplaced\ objects}{Total\ number\ of\ objects}.100 \tag{6}$$

k and *q* are, respectively, the number of clusters and true classes. *n* is the total number of objects (size of the dataset), n_{ij} is the number of objects that are from class *j* and clustered in cluster *i*. n_i and n_j are, respectively, the size of cluster *i* and class *j*.

Relative indices are different from the two aforementioned indices. They compare the results of different clustering algorithms or the same algorithm, yet with different parameters [10], [45].

3. RAT SWARM OPTIMIZER (RSO)

3.1 Inspiration

RSO [37] is a novel swarm intelligence technique inspired from two behaviors of rats in nature; chasing and fighting a prey. Black and brown rats are the two main species of rats. In general, rats show a social intelligence by nature. They contribute and help each other in different tasks. Rats live in groups and they are known by their aggressiveness in chasing and fighting prey, which is the fundamental motivation of the RSO algorithm.

In RSO, each rat represents a different solution. The RSO starts by initializing the set of solutions (rats) randomly and then evaluates them by an objective function, where the optimal solution is considered as the best rat $\overrightarrow{P_r}$ and so, the following processes are repeatedly executed a certain number of times (*T*), starting by firstly updating the position of each rat by the two behaviors chasing and fighting prey; secondly, the parameters are updated and any solution beyond the search space is adjusted and finally, the fitness of each rat is recalculated and the position of the best rat is updated if there is a better solution than $\overrightarrow{P_r}$. After completing that, the RSO returns the best solution $\overrightarrow{P_r}$. Algorithm 1 represents the pseudo-code of RSO.

3.2 Mathematical Model and Optimization Algorithm

The two behaviors of chasing and fighting the prey are modeled as follows.

3.2.1 Chasing the Prey

Rats' chasing is generally a social task. The best search agent is considered as the rat which has knowledge about the prey's location. The rest of the group will update their positions according to the best-rat position as follows [37]:

$$\overrightarrow{P_r} = A.\overrightarrow{P_i}(t) + C.(\overrightarrow{P_r}(t) - \overrightarrow{P_i}(t)) \tag{7}$$

where $\overrightarrow{P_i}(t)$ represents the position of the i^{th} rat (solution) and t represents the number of the current iteration. $\overrightarrow{P_i}(t)$ is the position of the best soon. A is calculated as follows:

$$A = R - t.\left(\frac{R}{max\ Iteration}\right) \tag{8}$$

R and *C* are random numbers, respectively, in [1,5] and [0,2]. *A* and *C* are two parameters for exploration and exploitation mechanisms.

$$R = rand(1,5) \tag{9}$$

$$C = rand(0,2) \tag{10}$$

3.2.2 Fighting the Prey

The fighting behavior is mathematically modeled as follows:

$$\overrightarrow{P_i}(t+1) = \left|\overrightarrow{P_i}(t) - \vec{P}\right| \tag{11}$$

where $\overrightarrow{P_i}(t+1)$ is the next position of rat number *i*.

A and *C* parameters are used to make balance between exploration and exploitation mechanisms. A small value of *A* (such as 1) and a moderate value of *C* will lead to emphasise exploitation. Other distant values may lead to emphasise exploration. The objective function used to evaluate results quality is the sum of intra-cluster distances which is defined as:

$$\sum_{C_i \in C} \sum_{x \in C_i} d^2(x, \mu_i) \tag{12}$$

μ_i is the center of the cluster *i* and $d^2(...)$ is the squared Euclidean distance.

Algorithm 1: RSO [37]

Parameter Initialization:
Initialize $\vec{R}$, $\vec{A}$ and $\vec{C}$ and set $t = 0$
Population Initialization:
Initialize the group of rats $P_i (i = 1,...,n)$
Calculate the fitness value of each rat
The best solution is assigned to $\overrightarrow{P_r}$
while ($t < T$) **do**
  for each rat **do**
    Update the position of the current rat by Equation (11)
  Update $\vec{R}$, $\vec{A}$ **and** $\vec{C}$ by Equations (9, 8 and 10)
  Adjust the rat if it goes beyond the search space
  Calculate the fitness value of each rat
  If the best solution of the current iteration is better than $\overrightarrow{P_r}$ **then**
    The position of $\overrightarrow{P_r}$ is updated to the position of the best solution
  $t \leftarrow t+1$
Return: $\overrightarrow{P_r}$

4. PROPOSED RSO-BASED CLUSTERING METHOD (RSOC)

In RSOC, the idea is to find the best cluster centers. Thus, each rat is represented by a vector of *k* cluster centers, where each cluster center is an object in a *d−dimentional* space (feature space). Hence, a solution can be represented in a *(k∗d)−dimensional* space as follows:

$$P_i = \left(\left(\mu_{i,1,1}, \mu_{i,1,2}, \dots, \mu_{i,1,d}\right), \left(\mu_{i,2,1}, \mu_{i,2,2}, \dots, \mu_{i,2,d}\right), \dots, \left(\mu_{i,k,1}, \mu_{i,k,2}, \dots, \mu_{i,k,d}\right)\right)$$

where, $\mu_{i,j,l}$ is the attribute number *l* of the center number $\boldsymbol{j}$ of the i^{th} rat.

The RSO process starts firstly by initializing each rat of the population by *k* random points from the dataset. The data is so clustered by each rat according to centers and each object is added to the cluster with the nearest center. After initializing parameters *A*, *C* and *R*, results are assessed by an objective function, where the best solution is saved in $\overrightarrow{P_r}$, then rats' positions are updated by Equation 11 and parameters *R*, *A* and *C* are so updated respectively by Equations (9, 8 and 10). If there is a rat beyond the search space, its position will be adjusted by reassigning the previous centers. The data is so clustered by each rat and the results are assessed by the objective function. If there is a better

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 08, No. 03, September 2022.

solution than $\overrightarrow{P_r}$, $\overrightarrow{P_r}$ is then updated to the position of the best solution. This process of rats' position updating continues until the end, where a max. number of iterations *T* are repeated. Finally, the data is clustered using the best cluster centers found ($\overrightarrow{P_r}$).

The pseudo-code (**Algorithm** 2) depicts the proposed RSOC.

Algorithm 2: RSOC

Parameter Initialization:
Number of clusters *k*, rats' group size, max. number of iterations *T* and the dataset
Population Initialization:
Data clustered with the best solution obtained $\overrightarrow{P_r}$
Initialize the group of rats $P_i (i = 1,...,n)$
Initialize $\vec{R}$, $\vec{A}$ and $\vec{C}$ by Equations (9, 8 and 10) and set $t = 0$
Cluster data by each rat
Assess results and the best solution is assigned to $\overrightarrow{P_r}$
while $(t < T)$ **do for**
  each rat **do**
    Update the position of the current rat by Equation (11)
    Cluster data by the current rat
    if *a solution is beyond the search space* **then**
      The current rat centers are not updated to the new centers.
      Update $\vec{R}$, $\vec{A}$ and $\vec{C}$ by Equations (9, 8 and 10)
      Calculate the fitness of the current solution by Equation (12)
    if *the best solution of the current iteration is better than* $\overrightarrow{P_r}$ **then**
      The position of $\overrightarrow{P_r}$ is updated to the position of the best solution
  $t \leftarrow t+1$
  Cluster the dataset by $\overrightarrow{P_r}$ and return the result
Return: $\overrightarrow{P_r}$ and data clustered with it

5. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, the proposed RSOC approach is applied to several real datasets and compared to other optimization algorithms. The results were measured for the first comparison by four measures: homogeneity: Equation (2), completeness: Equation (3), v-measure: Equation (4) and purity: Equation (5). Results are measured by error rate: Equation (6) for the second comparison. Table 1 details the utilized benchmark datasets, which are obtained from UCI Machine Learning Repository [49].

Table 1. Used datasets.

Dataset	Number of instances	Number of features	Number of classes
Iris	150	4	3
Ecoli	336	7	8
Glass	214	9	6
Heart	270	13	2
Cancer	683	10	2
Seeds	210	7	3
Wine	178	13	3
CMC	1473	9	3

5.1 Comparison with MVO

The RSOC here was compared with: Deferential Evolution (DE), Particle Swarm Optimization (PSO), Genetic Algorithm (GA) and Multi-verse Optimizer (MVO). The results were validated through four measures: homogeneity: Equation (2), completeness: Equation (3), v-measure: Equation (4) and purity: Equation (5). Parameters of algorithms compared with are the same mentioned in [35], since

the results are taken directly from [35]. For RSOC, maximum number of iterations and population size are the same as for MVO; 200 as max. number of iterations and 50 as population size. The results are gathered through 10 independent runs. The results are presented in Tables (2-6).

Table 2. Clustering results of Iris dataset.

	Homogeneity	Completeness	V-measure	Purity
DE	0.72778	0.75507	0.74096	0.86733
	(0.04379)	(0.04469)	(0.04293)	(0.03777)
PSO	0.65750	**0.82877**	0.72629	0.77133
	(0.07052)	**(0.09641)**	(0.02481)	(0.09270)
GA	0.60002	0.69056	0.64046	0.75333
	(0.09578)	(0.09905)	(0.09045)	(0.08433)
MVO	0.73642	0.74749	0.74191	0.88667
	(0.00000)	(0.00000)	(0.00000)	(0.00000)
RSOC	**0.74847**	0.76149	**0.75492**	**0.89200**
	(0.00635)	(0.00738)	**(0.00686)**	**(0.00281)**

Table 3. Clustering results of Ecoli dataset.

	Homogeneity	Completeness	V-measure	Purity
DE	0.43868	0.56485	0.49188	0.69235
	(0.10838)	(0.12341)	(0.11438)	(0.07287)
PSO	0.22629	**0.74693**	0.31740	0.57187
	(0.14762)	**(0.17063)**	(0.20040)	(0.09071)
GA	0.44054	0.52583	0.47512	0.67890
	(0.08253)	(0.08403)	(0.06779)	(0.06717)
MVO	0.50214	0.71637	0.58060	0.72508
	(0.13705)	(0.04119)	(0.10298)	(0.07459)
RSOC	**0.69627**	0.54324	**0.61021**	**0.81815**
	(0.01868)	(0.02423)	**(0.02162)**	**(0.01559)**

Table 4. Clustering results of Glass dataset.

	Homogeneity	Completeness	V-measure	Purity
DE	0.18996	0.46231	0.26717	0.45047
	(0.05362)	(0.08873)	(0.06913)	(0.03201)
PSO	0.17044	0.46871	0.24495	0.44206
	(0.07987)	(0.11835)	(0.10986)	(0.04295)
GA	0.24416	0.40213	0.30203	0.48972
	(0.04901)	(0.08900)	(0.05786)	(0.03763)
MVO	0.24341	**0.50376**	0.32666	0.47804
	(0.03544)	**(0.07557)**	(0.04368)	(0.02136)
RSOC	**0.36172**	0.43519	**0.39355**	**0.56028**
	(0.02865)	(0.07616)	**(0.04263)**	**(0.02611)**

Table 5. Clustering results of Heart dataset.

	Homogeneity	Completeness	V-measure	Purity
DE	0.13902	0.13881	0.13890	0.68815
	(0.11303)	(0.11186)	(0.11245)	(0.10174)
PSO	0.17086	0.20987	0.18129	0.70148
	(0.11114)	(0.08492)	(0.11056)	(0.10612)
GA	0.14584	0.15514	0.15021	0.70519
	(0.09743)	(0.10498)	(0.10090)	(0.08145)
MVO	**0.25875**	**0.25761**	**0.25816**	**0.78222**
	(0.06571)	**(0.06283)**	**(0.06432)**	**(0.05627)**
RSOC	0.01881	0.01944	0.01912	0.59074
	(0.00086)	(0.00092)	(0.00089)	(0.00195)

Table 6. Clustering results of Seeds dataset.

	Homogeneity	Completeness	V-measure	Purity
DE	0.55015	0.64305	0.58691	0.77048
	(0.10567)	(0.03752)	(0.06628)	(0.09162)
PSO	0.54263	0.68222	0.59593	0.76095
	(0.11405)	(0.05097)	(0.06504)	(0.11586)

GA	0.54015	0.61663	0.57184	0.76762
	(0.06536)	(0.05254)	(0.03513)	(0.08056)
MVO	0.61098	0.67855	0.63709	0.82810
	(0.09793)	(0.03824)	(0.05412)	(0.10025)
RSOC	**0.69394**	**0.69689**	**0.69541**	**0.89524**
	(0.00793)	**(0.00877)**	**(0.00835)**	**(0.00224)**

Tables (2-6) show the superiority of RSOC in most datasets. RSOC showed the best values outperforming all other techniques compared with in terms of homogeneity, v-measure and purity for all datasets, expect for Heart dataset, where it gave the worst values. MVO gave the best values on Heart dataset and on Glass dataset for completeness measure. PSO outperformed all other algorithms in terms of completeness for Iris and Ecoli datasets. However, for Seeds dataset, RSOC showed the best results in all measures. As presented in Tables (2-6), RSOC seems to find more homogeneous and pure clusters. To recapitulate, RSOC occupied the first place by outperforming other algorithms in 13 cases, 4 of which for homogeneity, 4 for purity, 4 for v-measure and one for completeness. MVO occupied the second place by outpassing other algorithms in 5 cases, 2 for completeness, one for homogeneity, one for v-measure and one for purity. At the third place, PSO outperformed other techniques in two cases for completeness.

5.2 Comparison with H-HHO

At the second comparison, RSOC was compared to a number of algorithms, namely: K-means++ (KM++) [52], Spectral, Agglomerative [53], DBSCAN [50], Genetic Algorithm (GA) [54], Particle Swarm Optimization (PSO) [55], Harmony Search (HS) [56], Krill Herd Algorithm (KHA) [57], Hybrid GA (H-GA) [50], Hybrid PSO (H-PSO) [51], H-KHA [50] and H-HHO [51]. Since the results were taken directly from [50]-[51], they are validated by error rate through five datasets: Iris, Wine, Cancer, CMC and Glass. Parameters of algorithms compared with are mentioned in [50]-[51]. Parameters of RSOC are set to be the same as for H-HHO, max number of iteration is set to (1000). Results are collected over 15 independent runs.

Table 7. Error-rate results.

	Criterion	Iris	Wine	Cancer	CMC	Glass	Rank
K-means	MEAN	21.467	32.388	42.388	55.470	46.154	
	BEST	10.660	29.775	39.865	54.660	42.262	12
	WORST	56.667	43.820	45.970	56.667	46.215	
KM++	MEAN	20.983	31.841	40.145	56.258	44.566	
	BEST	10.101	30.546	39.500	52.003	45.123	07
	WORST	54.274	43.534	44.965	57.001	45.250	
Spectral	MEAN	17.458	33.585	40.154	55.120	46.614	
	BEST	10.547	29.189	38.111	53.541	38.541	09
	WORST	55.541	43.137	44.685	54.044	51.991	
Agglomerative	MEAN	18.544	34.154	41.645	54.944	43.222	
	BEST	9.874	30.665	39.148	52.391	32.001	06
	WORST	48.397	42.688	46.699	57.487	52.140	
DBSCAN	MEAN	16.311	33.487	42.199	56.544	44.984	
	BEST	9.987	30.140	39.654	54.280	33.717	11
	WORST	43.111	42.009	44.021	56.654	51.123	
GA	MEAN	21.652	34.270	44.270	56.697	51.028	
	BEST	10.666	29.310	39.510	54.656	42.991	14
	WORST	43.333	47.753	47.753	57.296	56.075	
PSO	MEAN	15.867	32.051	43.051	55.899	46.262	
	BEST	10.667	29.775	40.775	54.101	43.925	10
	WORST	43.447	44.449	45.455	56.486	52.804	
HS	MEAN	21.054	32.568	42.054	56.001	43.054	
	BEST	10.509	29.865	40.111	55.430	41.162	08
	WORST	44.286	44.467	45.640	57.906	46.255	
KHA	MEAN	22.658	32.303	42.543	56.056	43.925	
	BEST	9.430	29.213	39.256	53.936	38.318	12

	WORST	42.548	47.191	47.191	56.999	50.476	
H-GA	MEAN	21.100	30.989	41.214	55.142	44.219	
	BEST	9.765	29.654	40.254	53.124	35.249	05
	WORST	44.667	44.001	46.214	56.214	51.985	
H-PSO	MEAN	15.800	30.871	42.125	54.204	51.617	
	BEST	9.666	29.775	39.775	53.201	41.589	04
	WORST	44.333	43.888	46.758	55.333	56.075	
H-KHA	MEAN	19.866	33.000	**39.012**	53.656	42.219	
	BEST	9.000	29.650	38.670	52.213	32.242	02
	WORST	43.333	42.134	44.154	54.333	51.420	
H-HHO	MEAN	20.866	33.564	39.470	54.109	44.002	
	BEST	9.332	29.653	39.119	53.165	34.242	03
	WORST	43.333	43.584	45.365	55.693	51.445	
RSOC	MEAN	**12.027**	**28.134**	45.737	**46.292**	**33.070**	
	BEST	12.027	28.134	45.737	46.208	31.587	**01**
	WORST	12.027	28.134	45.737	46.392	33.970	

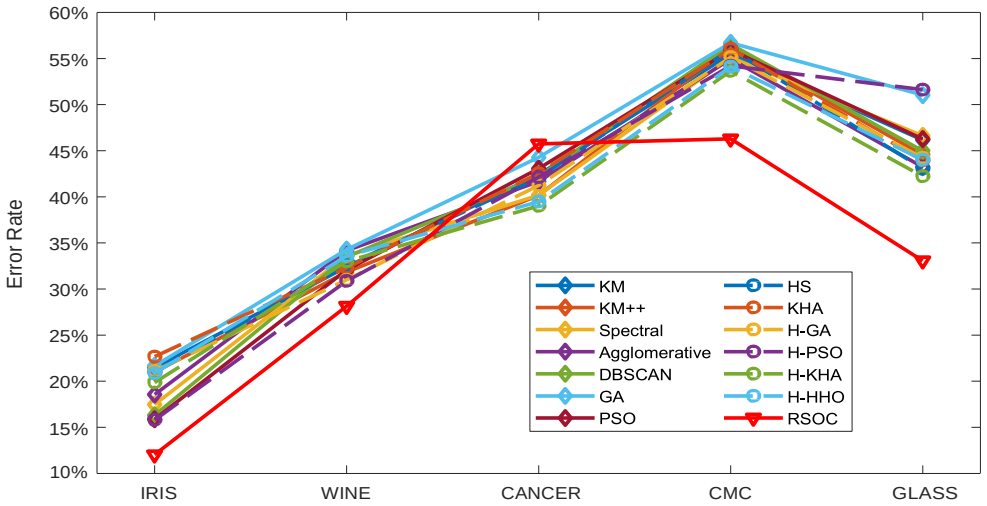


Figure 1. Visual comparison of error-rate results.

Table 8. Ranks of algorithms.

	Iris	Wine	Cancer	CMC	Glass	Sum
K-means	12	7	10	8	10	47
KM++	9	4	3	12	8	36
Spectral	5	12	4	6	12	39
Agglomerative	6	13	6	5	4	34
DBSCAN	4	10	9	13	9	45
GA	13	14	13	14	13	67
PSO	3	5	12	9	11	40
HS	10	8	7	10	2	37
KHA	14	6	11	11	5	47
H-GA	11	3	5	7	6	32
H-PSO	2	2	8	4	14	30
H-KHA	7	9	1	2	3	22
H-HHO	8	11	2	3	5	29
RSOC	1	1	14	1	1	18

Tables (7-8) and Figure 1 show impressive results, where RSOC ranked the first among other algorithms. It outperformed all other algorithms showing the least error rate on all datasets, expect for Cancer dataset, where it unexpectedly occupied the last place, which calls for no free lunch theorem (no algorithm is suitable

for all problems). Next to RSOC, comes H-KHA occupying the second place; first place on Cancer dataset and second place on CMC and Glass datasets. The third place went to H-HHO, which got the second place on Cancer dataset and the third place on CMC. The rest of algorithms are ordered as follows: H-PSO, H-GA, agglomerative clustering, k-means++, HS, spectral clustering, PSO, DBSCAN, k-means and KHA sharing the same rank and finally GA. RSOC showed a small deviation compared to other algorithms with CMC and Glass datasets and no deviation for the rest of datasets.

6. CONCLUSION AND FUTURE WORKS

In this work, we applied RSO technique for the problem of data clustering, where the number of clusters is known *a priori*. The proposed technique was compared to other algorithms and the quality of results was measured in terms of five measures in two comparisons: homogeneity, completeness, v-measure and purity for the first comparison and error rate for the second. Results and analysis showed the superiority of RSOC. However, this technique is still showing a weakness, such as on Heart and Cancer datasets, where it gave the worst values. As a future work, we will try to improve this technique and apply it to solve other problems, such as feature selection. We will also try to compare this metaheuristic to grey wolf optimizer, since they are very similar.

REFERENCES

[1] J. Han, M. Kamber and J. Pei, Data Mining: Concepts and Techniques, 3rd Edn., San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.

[2] A. M. Bagirov, N. Karmitsa and S. Taheri, Partitional Clustering *via* Nonsmooth Optimization: Clustering *via* Optimization, Springer Nature, 2020.

[3] P. Berkhin, "A Survey of Clustering Data Mining Techniques," Proc. of Grouping Multidimensional Data, pp. 25–71, Springer, 2006.

[4] B. Everitt, S. Landau, M. Leese and D. Stahl, Cluster Analysis, ser. Wiley Series in Probability and Statistics, Wiley, [Online], Available: https://books.google.dz/books?id=WSayDAEACAAJ, 2011.

[5] J. Hartigan, Clustering Algorithms, John Wiley and Sons, New York, 1975.

[6] K. Krippendorff, "Clustering," Book Chapter, Multivariate Techniques in Human Communication Research, pp. 259-308, Elsevier, 1980.

[7] K. Bailey, "Cluster Analysis," Book Chapter, Sociological Methodology, pp. 59-128, DOI: 10.2307/270894, 1975.

[8] A. K. Jain and R. C. Dubes, Algorithms for Clustering Data, USA: Prentice-Hall, Inc., 1988.

[9] B. Mirkin, Clustering for Data Mining: A Data Recovery Approach, DOI:10.1201/9781420034912, 2005.

[10] R. Xu and D. Wunsch, Clustering, vol. 10, John Wiley & Sons, 2008.

[11] A. Naik and S. C. Satapathy, "Past Present Future: A New Human-based Algorithm for Stochastic Optimization," Soft Computing, vol. 25, no. 20, pp. 12 915–12 976, 2021.

[12] M. Dorigo, V. Maniezzo and A. Colorni, "Ant System: Optimization by a Colony of Cooperating Agents," IEEE Trans. on Systems, Man and Cybernetics, Part B (Cybernetics), vol. 26, no. 1, pp. 29–41, 1996.

[13] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," Proc. of the International Conference on Neural Networks (ICNN'95), vol. 4, , pp. 1942–1948, 1995.

[14] S. Z. Selim and K. Alsultan, "A Simulated Annealing Algorithm for the Clustering Problem," Pattern Recognition, vol. 24, no. 10, pp. 1003–1008, 1991.

[15] K. S. Al-Sultan, "A Tabu Search Approach to the Clustering Problem," Pattern Recognition, vol. 28, no. 9, pp. 1443–1451, 1995.

[16] F. Glover, "Future Paths for Integer Programming and Links to Artificial Intelligence," Computers & Operations Research, vol. 13, no. 5, pp. 533–549, 1986.

[17] F. Glover, "Tabu Search—Part i," ORSA Journal on Computing, vol. 1, no. 3, pp. 190–206, 1989.

[18] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, 1st Edn., USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[19] J. H. Holland, Adaptation in Natural and Artificial Systems, Ann Arbor, MI: University of Michigan Press, 1975, 2nd Edition, 1992.

[20] M. C. Cowgill, R. J. Harvey and L. T. Watson, "A Genetic Algorithm Approach to Cluster Analysis," Computers & Mathematics with Applications, vol. 37, no. 7, pp. 99–108, 1999.

[21] E. Falkenauer, Genetic Algorithms and Grouping Problems, John Wiley & Sons, Inc., 1998.

[22] E. R. Hruschka, R. J. Campello, A. A. Freitas et al., "A Survey of Evolutionary Algorithms for Clustering," IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews), vol. 39, no. 2, pp. 133–155, 2009.

[23] P. Shelokar, V. K. Jayaraman and B. D. Kulkarni, "An Ant Colony Approach for Clustering," Analytica Chimica Acta, vol. 509, no. 2, pp. 187–195, 2004.

[24] J. Ji, W. Pang, Y. Zheng, Z. Wang and Z. Ma, "A Novel Artificial Bee Colony Based Clustering Algorithm for Categorical Data," PloS One, vol. 10, no. 5, p. e0127125, 2015.

[25] D. Karaboga and B. Basturk, "An Artificial Bee Colony (ABC) Algorithm for Numeric Function Optimization," Proc. of the IEEE Swarm Intelligence Symposium, pp. 181–184, Indianapolis, USA, 2006.

[26] D. Van der Merwe and A. P. Engelbrecht, "Data Clustering Using Particle Swarm Optimization," Proc. of the IEEE Congress on Evolutionary Computation (CEC'03), vol. 1, pp. 215–220, 2003.

[27] Y.-T. Kao, E. Zahara and I.-W. Kao, "A Hybridized Approach to Data Clustering," Expert Systems with Applications, vol. 34, no. 3, pp. 1754–1762, 2008.

[28] T. Cura, "A Particle Swarm Optimization Approach to Clustering," Expert Systems with Applications, vol. 39, no. 1, pp. 1582–1588, 2012.

[29] X. Zhang, Q. Lin, W. Mau, Z. Dou and G. Liu, "Hybrid Particle Swarm and Grey Wolf Optimizer and Its Application to Clustering Optimization, " Applied Soft Computing, vol. 101, p. 107061, 2021.

[30] S. Mirjalili, S. M. Mirjalili and A. Lewis, "Grey Wolf Optimizer," Advances in Engineering Software, vol. 69, pp. 46–61, 2014.

[31] V. Kumar, J. K. Chhabra and D. Kumar, "Grey Wolf Algorithm-based Clustering Technique," Journal of Intelligent Systems, vol. 26, no. 1, pp. 153–168, 2017.

[32] N. Kushwaha, M. Pant, S. Kant and V. Jain, "Magnetic Optimization Algorithm for Data Clustering," Pattern Recognition Letters, vol. 115, pp. 59-65, [Online], Available: 10.1016/j.patrec.2017.10.031, 2018.

[33] H. A. Abdulwahab, A. Noraziah, A. A. Alsewari and S. Q. Salih, "An Enhanced Version of Black Hole Algorithm *via* Levy Flight for Optimization and Data Clustering Problems," IEEE Access, vol. 7, pp. 142085-142096, DOI: 10.1109/ACCESS.2019.2937021, 2019.

[34] I. Aljarah, M. Mafarja, A. A. Heidari, H. Faris and S. Mirjalili, "Clustering Analysis Using a Novel Locality-informed Grey Wolf-inspired Clustering Approach," Knowledge and Information Systems, vol. 62, no. 2, pp. 507–539, 2020.

[35] I. Aljarah, M. Mafarja, A. A. Heidari, H. Faris and S. Mirjalili, "Multi-verse Optimizer: Theory, Literature Review and Application in Data Clustering," Nature-inspired Optimizers, vol. 811, pp. 123–141, 2020.

[36] S. Mirjalili, S. M. Mirjalili and A. Hatamlou, "Multi-verse Optimizer: A Nature-inspired Algorithm for Global Optimization," Neural Computing and Applications, vol. 27, no. 2, pp. 495–513, 2016.

[37] G. Dhiman, M. Garg, A. Nagar, V. Kumar and M. Dehghani, "A Novel Algorithm for Global Optimization: Rat Swarm Optimizer," Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 8, pp. 8457–8482, 2021.

[38] M. Dhas and N. Singh, "Blood Cell Image Denoising Based on Tunicate Rat Swarm Optimization with Median Filter," Evolutionary Computing and Mobile Sustainable Networks, vol. 116, pp. 33–45, 2022.

[39] A. Tamilarasan, A. Renugambal and V. Dharanendran, "Parametric Estimation for AWJ Cutting of TI-6AL-4V Alloy Using Rat Swarm Optimization Algorithm," Materials and Manufacturing Processes, pp. 1–11, DOI: 10.1080/10426914.2022.2065011, 2022.

[40] M. Eslami, E. Akbari, S. T. Seyed Sadr and B. Ibrahim, "A Novel Hybrid Algorithm Based on Rat Swarm Optimization and Pattern Search for Parameter Extraction of Solar Photovoltaic Models," Energy Science and Engineering, DOI: 10.1002/ese3.1160, 2022.

[41] R. Ghadge and S. Prakash, "Investigation and Prediction of Hybrid Composite Leaf Spring Using Deep Neural Network Based Rat Swarm Optimization," Mechanics Based Design of Structures and Machines, pp. 1–30, DOI: 10.1080/15397734.2021.1972309, 2021.

[42] A. Vasantharaj, P. Rani, S. Huque, K. Raghuram, R. Ganeshkumar and S. Shafi, "Automated Brain Imaging Diagnosis and Classification Model Using Rat Swarm Optimization with Deep Learning Based Capsule Network," International Journal of Image and Graphics, p. 2240001, DOI: 10.1142/S0219467822400010, 2021.

[43] G. Gan, C. Ma and J. Wu, Data Clustering: Theory, Algorithms and Applications, SIAM, 2020.

[44] M. Halkidi, Y. Batistakis and M. Varzigiannis, "Cluster Validity Methods: Part I," ACM Sigmod Record, vol. 31, no. 2, pp. 40-45, 2002.

[45] M. Halkidi, Y. Batistakis and M. Vazirgiannis, "Clustering Validity Checking Methods: Part II,"ACM Sigmod Record, vol. 31, no. 3, pp. 19–27, 2002.

[46] J. Oyelade et al., "Data Clustering: Algorithms and Its Applications," Proc. of the 19th IEEE Int. Conf. on Computational Science and Its Applications (ICCSA), pp. 71–81, St. Petersburg, Russia, 2019.

[47] R. Zafarani, M. A. Abbasi and H. Liu, Social Media Mining: An Introduction, Cambridge University Press, ISBN-10: 1107018854, 2014.

[48] J.-O. Palacio-Niño and F. Berzal, "Evaluation Metrics for Unsupervised Learning Algorithms," arXiv preprint arXiv:1905.05667, 2019.

[49] D. Dua and C. Graff, "UCI Machine Learning Repository," [Online], Available: http://archive.ics.uci.edu/ml, 2017.

[50] L. Abualigah, A. Khader, E. Hanandeh and A. Gandomi, "A Novel Hybridization Strategy for Krill Herd Algorithm Applied to Clustering Techniques", Applied Soft Computing, vol. 60, pp. 423-435, 2017.

[51] L. Abualigah et al., "Hybrid Harris Hawks Optimization with Differential Evolution for Data Clustering,"

Metaheuristics in Machine Learning: Theory and Applications, vol. 967, pp. 267-299, 2021.

[52] D. Arthur and V. Sergei, "K-Means++: The Advantages of Careful Seeding," Proc. of the 18th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA '07), pp. 1027-1035, 2007.

[53] I. Davidson and S.S. Ravi, "Agglomerative Hierarchical Clustering with Constraints: Theoretical and Empirical Results," Knowledge Discovery in Databases: PKDD 2005, pp. 59-70, [Online], Available: 10.1007/11564126_11, 2005.

[54] U. Maulik and S. Bandyopadhyay, "Genetic Algorithm-based Clustering Technique," Pattern Recognition, vol. 33, no. 9, DOI: 10.1016/S0031-3203(99)00137-5, 2000.

[55] S. Rana, S. Jasola and R. Kumar, "A Review on Particle Swarm Optimization Algorithms and Their Applications to Data Clustering," Artificial Intelligence Review, vol. 35, no. 3, pp. 211-222, 2010.

[56] O. Alia, M. Al-Betar, R. Mandava and A. Khader, "Data Clustering Using Harmony Search Algorithm," Proc. of Int. Conf. on Swarm, Evolutionary and Memetic Computing (SEMCCO 2011), pp. 79-88, DOI: 10.1007/978-3- 642-27242-4_10, 2011.

[57] L. Abualigah, A. Tajudin Khader, M. Azmi AlBetar and E. Said Hanandeh, "A New Hybridization Strategy for Krill Herd Algorithm and Harmony Search Algorithm Applied to Improve the Data Clustering," Proc. of the 1st EAI Int. Conf. on Computer Science and Engineering, DOI: 10.4108/eai.27-2-2017.152255, 2017.

ملخص البحث:

تعدّ عملية الأمْثَلَة باستخدام "سِرب الجرذان" من أحدث تطبيقات الأمْثَلَة اعتماداً على خوارزميات يتمّ استلهامها من سلوك أسراب الجرذان المتمثّل في مطاردة الضّحية والانقضاض عليها.

في هذا البحث، نعمل على تطبيق نظام أمْثَلَة يعتمد سلوك "سِرب الجرذان" على مشكلة هي من أبرز التّحدّيات تتمثّل في عَنْقَدة البيانات. وتعمل قدرة هذا النّظام على البحث على إيجاد أفضل مراكز عناقيد البيانات.

وقد جرى فحص النّظام المقترح بناءً على عدّة علامات مرجعية ومقارنته مع عدد من الأنظمة الأخرى المستخدمة في عَنْقَدة البيانات المعتمدة على خوارزميات قوية معروفة جيداً. وتمّ تقييم النتائج بواسطة حزمة من المقاييس، مثل: التّجانس، والاكتمال، ومقياس (v)، والنّقاء، ومعدّل الخطأ. وقد أسفرت نتائج الحسابات على استنتاجاتٍ مشجّعة، ممّا أثبت فعالية التّقنية المقترحة وتفوّقها بشكلٍ لافت على التّقنيات الأخرى المستخدمة في عَنْقَدة البيانات.

المجلة الأردنية للحاسوب وتكنولوجيا المعلومات (JJCIT) مجلة علمية عالمية متخصصة محكمة تنشر الأوراق البحثية الأصيلة عالية المستوى في جميع الجوانب والتقنيات المتعلقة بمجالات تكنولوجيا وهندسة الحاسوب والاتصالات وتكنولوجيا المعلومات.

تحتضن وتنشر جامعة الأميرة سمية للتكنولوجيا (PSUT) المجلة الأردنية للحاسوب وتكنولوجيا المعلومات، وهي تصدر بدعم من صندوق دعم البحث العلمي في الأردن. وللباحثين الحق في قراءة كامل نصوص الأوراق البحثية المنشورة في المجلة وطباعتها وتوزيعها والبحث عنها وتنزيلها وتصويرها والوصول إليها. وتسمح المجلة بالنسخ من الأوراق المنشورة، لكن مع الإشارة إلى المصدر.

الأهداف والمجال

تهدف المجلة الأردنية للحاسوب وتكنولوجيا المعلومات (JJCIT) إلى نشر آخر التطورات في شكل أوراق بحثية أصيلة وبحوث مراجعة في جميع المجالات المتعلقة بالاتصالات وهندسة الحاسوب وتكنولوجيا المعلومات وجعلها متاحة للباحثين في شتى أرجاء العالم. وتركز المجلة على موضوعات تشمل على سبيل المثال لا الحصر: هندسة الحاسوب وشبكات الاتصالات وعلوم الحاسوب ونظم المعلومات وتكنولوجيا المعلومات وتطبيقاتها.

الفهرسة

المجلة الأردنية للحاسوب وتكنولوجيا المعلومات مفهرسة في كل من:

فريق دعم هيئة التحرير

المحرر اللغوي

حيدر المومني

ادخال البيانات وسكرتير هيئة التحرير

إياد الكوز

جميع الأوراق البحثية في هذا العدد مُتاحة للوصول المفتوح، وموزعة تحت أحكام وشروط ترخيص [Creative Commons Attribution] (http://creativecommons.org/licenses/by/4.0/)

عنوان المجلة

الموقع الإلكتروني: www.jjcit.org
البريد الإلكتروني: jjcit@psut.edu.jo
العنوان: جامعة الاميرة سمية للتكنولوجيا، شارع خليل الساكت، الجبيهة، عمان، الأردن.
صندوق بريد: 1438 عمان 11941 الأردن
هاتف: 5359949-6-962+
فاكس: 7295534-6-962+

جامعة الأميرة سميّة للتكنولوجيا
Princess Sumaya University for Technology

صندوق دعم البحث العلمي والابتكار
Scientific Research and Innovation Support Fund

المجلة الأردنية للحاسوب وتكنولوجيا المعلومات

أيلول ٢٠٢٢ المجلد ٨ العدد ٣
ISSN 2415 - 1076 (Online)
ISSN 2413 - 9351 (Print)

JJCIT

الصفحات	عنوان البحث

www.jjcit.org
jjcit@psut.edu.jo

مجلة علمية عالمية متخصصة تصدر
بدعم من صندوق دعم البحث العلمي والابتكار