In the Name of God

Journal of

Information Systems & Telecommunication

Vol. 10, No.2, April-June 2022, Serial Number 38

Research Institute for Information and Communication Technology
Iranian Association of Information and Communication Technology
Affiliated to: Academic Center for Education, Culture and Research (ACECR)

Manager-in-Charge: Dr. Habibollah Asghari, ACECR, Iran
Editor-in-Chief: Dr**.** Masoud Shafiee, Amir Kabir University of Technology, Iran

Executive Editor: Dr. Fatemeh Kheirkhah
Executive Manager: Shirin Gilaki
Executive Assistants: Mahdokht Ghahari, Ali Mokhtarani, Ali BoozarPoor
Print ISSN: 2322-1437
Online ISSN: 2345-2773
Publication License: 91/13216
Editorial Office Address: No.5, Saeedi Alley, Kalej Intersection., Enghelab Ave., Tehran, Iran,
P.O.Box: 13145-799 Tel: (+9821) 88930150 Fax: (+9821) 88930157
E-mail: info@jist.ir , infojist@gmail.com
URL: www.jist.ir

Indexed by:

- SCOPUS	www. Scopus.com
- Index Copernicus International	www.indexcopernicus.com
- Islamic World Science Citation Center (ISC)	www.isc.gov.ir
- Directory of open Access Journals	www.Doaj.org
- Scientific Information Database (SID)	www.sid.ir
- Regional Information Center for Science and Technology (RICeST)	www.ricest.ac.ir
- Iranian Magazines Databases	www.magiran.com

Publisher:
Iranian Academic Center for Education, Culture and Research **(ACECR)**

This Journal is published under scientific support of
Advanced Information Systems (AIS) Research Group and
Telecommunication Research Group, ICTRC

Acknowledgement

JIST Editorial-Board would like to gratefully appreciate the following distinguished referees for spending their valuable time and expertise in reviewing the manuscripts and their constructive suggestions, which had a great impact on the enhancement of this issue of the JIST Journal.

(A-Z)

- Abdolvand, Neda, Azahra University, Tehran, Iran
- Afsharirad, Majid, Kharazmi University,Tehran, Iran
- Ahmadifar, Hamidreza, University of Guilan, Rasht, Iran
- Ahmadizad, Arman, University of Kurdistan, Kurdistan, Iran
- Alizadeh Noughabi, Havva, Islamic Azad University of Gonabad, Iran
- Bayat, Peyman, Islamic Azad University, Rasht , Iran
- Deldari, Arash, University of Torbat Heydariyeh, Khorasan Razavi, Iran,
- Ebadati, Omid Mahdi, Kharazmi University,Tehran, Iran
- Farsi, Hassan, University of Birjand, South Khorasan, Iran
- Farsijani, Hassan, Shahid Beheshti University, Tehran, Iran
- Ghasemzadeh, Mohammad, Yazd University, Yazd, Iran
- Ghaffari, Ali, Islamic Azad University, Tabriz Branch, Iran
- Kheirkhah, Fatemeh, ACECR, Tehran, Iran
- Kashef, Seyed Sadra, Urmia University, Urmia, Iran
- Mahdieh, omid, University of Zanjan, Zanjan, Iran
- Mahmoodzadeh, Azar, Islamic Azad University, Shiraz, Iran
- Moradi, Gholamreza, Amirkabir University,Tehran, Iran
- Mirzaei, Abbas, Islamic Azad University, Ardabil, Iran
- Movahedi, Zeynab, University of Science and Technology, Tehran, Iran
- Pirgazi, Jamshid, Mazandaran University of Science and Technology, Mazandaran, Iran
- Qingliang, Zhao, Northwestern University, Evanston, United State
- Rasi, Habib, University of Technology, Shiraz, Iran
- Ramezani, Reza, Isfahan University, Isfahan, Iran
- R Ravinder Reddy, Chaitanya Bharathi Institute Technology, Hyderabad, India
- Rizal, Achmad, Telkom University, Bandung , Indonesia
- Rachid, Saadane, CSEE, Essex University, United Kingdom
- Raghavendra, B.K, BGSIT University, Karnataka, India
- Sharma, Manik, DVA University, Panjab, India
- Shahbahrami, Asadollah, Guilan University, Guilan, Iran
- Sarrafzadeh, Abdolhossein, Unitec Institute of Technology, Auckland, New Zealand
- Tayefeh Mahmoodi, Maryam, Research Institute for Information and Communication Technology, Tehran, Iran
- Taghizadeh Dehkordi, Maryam, Shahr-e Kord University, Chaharmahal and Bakhtiari, Iran
- V P Chandra Sekhar Yadav Gampa, K L University, Andhra, India
- Yaghoobi, Kaebeh, Ale Taha Institute of Higher Education, Tehran, Iran

Table of Contents

Journal of Information Systems and Telecommunication
Vol.10, No.2, April-June 2022, 80-88

http://jist.acecr.org
ISSN 2322-1437 / ESSN:2345-2773

A Novel Effort Estimation Approach for Migration of SOA Applications to Microservices

Vinay Raj[1*], Sadam Ravichandra[2]

[1].BVRIT Hyderabad, College of Engineering for Women, Hyderabad, India.
[2].National Institute of Technology Warangal, Telangana, India.

Received: 20 Aug 2020 / Revised: 27 Jul 2021/ Accepted: 05 Dec 2021
DOI:

Abstract

Microservices architecture's popularity is rapidly growing as it eases the design of enterprise applications by allowing independent development and deployment of services. Due to this paradigm shift in software development, many existing Service Oriented Architecture (SOA) applications are being migrated to microservices. Estimating the effort required for migration is a key challenge as it helps the architects in better planning and execution of the migration process. Since the designing style and deployment environments are different for each service, existing effort estimation models in the literature are not ideal for microservice architecture. To estimate the effort required for migrating SOA application to microservices, we propose a new effort estimation model called ***Service Points***. We define a formal model called service graph which represents the components of the service based architectures and their interactions among the services. Service graph provides the information required for the estimation process. We recast the use case points method and model it to become suitable for microservices architecture. We have updated the technical and environmental factors used for the effort estimation. The proposed approach is demonstrated by estimating the migration effort for a standard SOA based web application. The proposed model is compatible with the design principles of microservices and provides a systematic and formal way of estimating the effort. It helps software architects in better planning and execution of the migration process.

Keywords: Service Oriented Architecture; Microservices; Migration; Service Graph; Effort Estimation.

1- Introduction

Distributed systems have evolved rapidly beginning with the monolithic style of designing applications. Monolithic application has a large codebase, deployed as a single unit and the components of the application are highly coupled. Monolithic architecture has a limitation in the size and complexity of the application.

With the increase in the complexity of enterprise applications, business requirements, and the need for designing distributed applications has led to the evolution of SOA [1]. Service oriented architecture (SOA) has been widely used in designing large enterprise applications in the last two decades. It has mainly emerged to overcome the scalability and deployment challenges of monolithic applications. SOA is a style of designing applications where all the components in the system are designed as services. A service is a reusable software code that performs various business tasks that can be simple or complex based on the business requirements. SOA is mainly used in the integration of multiple software components using the Enterprise Service Bus (ESB) as the communication channel [2]. ESB is the backbone of SOA which helps in providing the features of the middleware system. ESB acts as a mediator between the service requester and provider and provides a platform for high performance and scalability. SOA gained more popularity with the evolution of web services which is the popular implementation of SOA concepts. Web services are also services that can be designed, accessed, and discovered over the internet using communication protocols such as XML based Simple Object Access Protocol (SOAP) and Web Service Description Language (WSDL). Web services use HTTP and Representational State Transfer (REST) protocols for the transfer of messages through the internet. The web services architecture comprises three main components namely service provider, service consumer, and service registry. A single web service can be used by multiple clients at the same time and can be easily deployed. Though SOA has gained huge demand in designing applications, it exhibits few design and deployment challenges [3]. Many changes and updates occur in large enterprise applications [25] and when there is a need to update a particular service, due to the

🖂 **Corresponding Author**
rvinay1@student.nitw.ac.in

dependency it has on other services and the existence of tight coupling with ESB, it requires to redeploy multiple components for a change in a single service. Deploying multiple services at a time leads SOA to the monolithic style of deployment and it impacts the business [4]. Additionally, with the increase in ever-changing business requirements, few services in SOA are tending towards monolithic in size making the application complex and difficult to maintain. Scaling such monolithic applications is a bottleneck as SOA follows centralized governance [5]. Services that are overloaded can be scaled horizontally by making multiple copies of the same service but the hardware cost increases. Further, web services use complex and heavyweight protocols such as SOAP for the exchange of messages between the services [27].

To over these challenges in existing architectures, microservices emerged as a new style of designing applications using cloud-based containers for deployment [26]. It is a style of designing applications where each service is a small, loosely coupled, scalable and reusable service that can be designed and deployed independently [6]. Each service should perform only one task and should have its own database and independent deployment architecture. Microservices uses communication protocols like HTTP/REST and JSON for data exchange between the services. Unlike SOA, microservices can be deployed independently as there is no centralized governance and no dependency on middleware technologies. It is very easy to scale on-demand microservices with the use of cloud-based containers. Microservices architecture suits well with the DevOps style as every task is to be broken into small units and complete SDLC is to be done independently [7]. DevOps and agile methodologies require the fast design of applications and deployment to production.

With the various benefits of microservices, software architects have started migrating their existing legacy applications to microservices architecture [8]. Many companies including Netflix, Amazon, and Twitter have started building their new applications with this style of architecture [9]. As microservices has emerged recently, there is a huge demand in both industry and academia to explore the tools, technologies, and programming languages used in this architecture. However, some of the software architects are in chaos whether to migrate to this new style or not as they are unaware of the pros and cons of using microservices. The major challenge is estimating the effort required to migrate the existing applications to microservices [8][10].

Effort estimation helps software architects in the proper execution and management of the project. Effective estimation helps in proper scheduling of the software engineering activities. Software effort is given by the formula effort = people * time [11]. It has to be done during the early stage of the application design as it gives insights on the effort and cost required to complete the application. Moreover, estimating the accurate effort required for the migration process is a challenging task. Underestimation and overestimation of the effort required may lead to serious project management issues. Software effort estimation techniques are divided into four types namely, empirical, regression, theory-based, and machine learning techniques based estimation [12]. Empirical way of estimating is very popular as it gives a clear picture of the effort required numerically and few of the models include function point, use case point, and analogy based techniques. The function point and COCOMO model fail to estimate the effort and cost required to design the application [13]. Parametric and non-parametric forecasting models are used in regression approaches. Multiple Linear Regression (MLR), Stepwise Regression (SR), Poisson Regression, Standard Regression, Ordinary Least Squares (OLS), and Stepwise Analysis of Variance are some of the most used regression approaches. Theory-based approaches are based on theoretical concepts that characterize certain parts of software development processes. Machine Learning based approaches for estimating software effort includes Artificial Neural Networks, Classification and Regression Tree, Case-based Reasoning, Genetic Algorithm, Genetic programming, and Rule Induction. Moreover, these techniques are not suitable for measuring the effort for service-based systems as they are designed for procedural object-oriented systems.

Use Case Points (UCP) is a commonly used technique because of its simplicity, fastness, and accuracy to a certain extent [14]. UCP approach is based on the use case diagrams for calculating the effort. Many variations and enhancements have been published in the literature to improve the accuracy of the approach [15][16][17]. Though the use case point approach is based on the use case diagrams of object-oriented concepts, attempts have been made for estimating the effort for service-oriented architectures [18]. All the traditional approaches available for effort estimation cannot be used directly for service-based systems. Approaches need to be modified and extended to cope with these service-based systems like service oriented architecture and microservices architecture [19].

To the best of our knowledge, there has been no work or very little work done in estimating the effort required for migration of service oriented architectures to microservices architecture. In this paper, we attempt to propose an approach for effort estimation by recasting the existing use case point model by enhancing it to suit appropriately for microservices. Generally, effort estimation requires knowing about the system before the design phase which is difficult. In our work, we use the service graph representation of the microservices application which is generated by the migration approach

[20] and it gives detailed information about the number of services and dependency it has on other services.

The remaining paper is organized as follows. The types of services involved in the migration process are discussed in section 2. The approach for effort estimation is presented in section 3. Evaluation of the proposed approach using a case study application is presented in section 4. Section 5 concludes the paper.

2- Types of Services Involved During the Migration Process

To migrate SOA based applications to a microservices architecture, the monolithic services need to be broken into small and independent services. However, there may exist few services in SOA based application which perform a single business task and can be directly considered as microservices. For systematic estimation of the effort, business services are classified into available, migrated, new, or composed services [21]. However, there exist many other types of services that are involved in achieving the business requirements such as utility services, process services, proxy services, integration services and, suspended services, etc. Here, we discuss the significance of each service in the migration of SOA to microservices architecture.

Available service: Services that can be used directly in the new architecture are treated as available services. Service which does a single business task and is independent of other services can be directly considered as microservice. It requires no development effort and hence it is considered as available service.

Migrated service: Service which is extracted from legacy applications and generated by applying different migration strategies is considered as migrated service. Here, the services in SOA which are partitioned to form microservices will be considered as migrated service. These services require an effort for redesigning the new application. The difference between available service and migrated service is available service can be directly used as microservice whereas migrated service requires effort for transforming itself to a new microservice.

New service: Service which is built from scratch and required for achieving the business needs is considered a new service. It requires effort and it is very easy to calculate the effort for new service. However, as both SOA and microservices architectures are service-based systems, no new services will be required while migration from SOA to microservices. Therefore, we will not consider this kind of service in effort estimation.

Composed service: Service which is formed by combining one or more services is considered as composed service. By the definition of microservices, each service should perform only a single task and independent from other services. Therefore, there will be no composed services in the new architecture.

It is inferred from the above that in the effort estimation of the migration process, only the migrated services need to be considered. So the proposed model considers only the migrated services in the effort estimation.

3- Effort Estimation Approach

3-1- Service Graph

Graph theory has been widely used in solving many complex problems in software engineering as the flow of messages and dependency between the software components can be graphically represented. As services are the software components in SOA based applications, we develop a new graph called service graph (SG) to extract the candidate microservices. We start by introducing the concept of service graph which plays a fundamental role in our proposed approach.

Service graph (SG) is a regular graph generated for the visual analysis of communication and dependency between the services of an SOA application. The service graph is the simplest representation of the number of services and the interactions among those services. The generalized form of any given SOA based application as a service graph is shown in Fig 1.

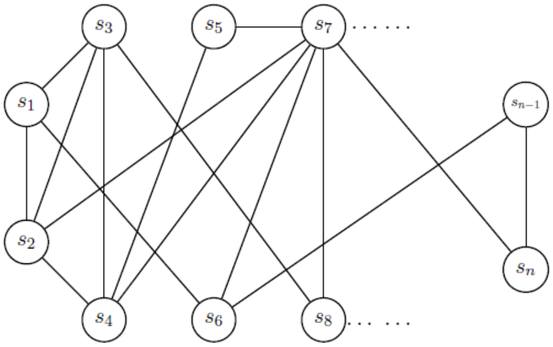


Figure 1: Formal Representation of SOA application

3-1-1-Service Definition

Let a graph G(V, E) be a service graph with **n** nodes, where the nodes of the graph represent a set of services in the application, and edges between the nodes represent the interactions or dependency each service has with other services in the application.

Let V={s_1,s_2,s_3,...} be the nodes of the service graph where s_1,s_2,s_3,… are services and E= {(s_1,s_2), (s_1,s_3), (s_2,s_4), ...} be the edges between the nodes which represent the dependency between the services. A service can be represented as a set of coordinating and interacting processes as defined in Eq. (1).

$$S_i = < P_1^i, P_2^i, P_3^i, \ldots.., P_n^i, \Lambda > \quad (1)$$

where S_i is the logical service instance, P_k^i indicates k^th process implementing logical service functionality f_i through the programmatic interface I_i and Λ represents network communication function between individual processes [22].

3-2-Proposed Approach

Our approach is stimulated from the use case points model of effort estimation. The use case point method depends on the use case diagram and our model depends on the service graph as we are estimating the effort for service-based architectures. The service graph is a blueprint for the application to be designed and it gives complete information regarding the number of services and complexity of the services based on the dependencies on other services. Similar to the use case point method, we propose a service point (SP) model to estimate the effort required for migration to microservices. We classify the services and then calculate the weights and points using the classification of the services. Technical and environmental factors are two important factors that play a major role in effort estimation. The factors accessed for the existing use case point method does not suit well for microservices architecture. Therefore, we have updated the technical and environmental factors considering the principles of service-oriented systems. The steps for effort estimation using the service point technique is illustrated in Fig. 2.

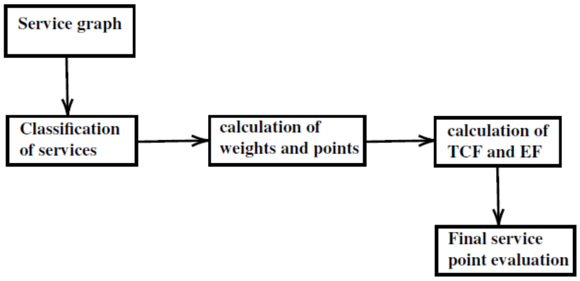

Service graph
Classification of services
calculation of weights and points
calculation of TCF and EF
Final service point evaluation

Figure 2: Service point calculation steps

3-2-1-Classification of Services

The first step of the service point approach is to classify the services based on the interactions it has with other services. Unlike the use case point, we don't have actors here. So, we consider the dependencies each service has on other services and classify them as simple, average, and complex. The service graph helps in the identification of services and their dependencies. A service is classified as simple if it interacts with less than four services, average if it interacts with less than eight services, and service is treated as complex if it interacts with more than or equal to eight services. The number of interacting services for a particular service helps in determining the service complexity. Based on the complexity, different weights are assigned to each service which is used in the calculation of service weights. The classification of services and the weights assigned are given in Table 1.

Table 1: Classification of services with weights

Service Complexity	*Number of interacting services*	*Weight*
Simple	Less than or equal to 3	5
Average	4 to 7	10
Complex	More than 7	15

3-2-2-Calculation of weights and points

The next step is to calculate the unadjusted service points based on the weights assigned in Table 1. It is calculated by summation of number of services of each type multiplied by weight assigned to corresponding service type. Unadjusted Service Points (USP) is calculated as shown in Eq. (2).

$$USP = \sum_{i=1}^{3} S_i \times W_i \qquad (2)$$

Where S_i is the number of services of type i and W_i is the corresponding weight of the service of type i where i={simple, average, complex}.

3-2-3-Technical and Environmental factors

We calculated the unadjusted service point value from the Eq. (2) and the final value of service point depends on technical and environmental factors. The 21 factors [23] relate to the factors which contribute to the complexity and the efficiency of the system. However, most of the factors included in existing works presented in the literature are not suitable for both service oriented architecture and microservices. Therefore, we have removed few factors and added new factors relevant to microservices architecture.

Each factor has a value assigned between 0 and 5 depending on the importance and impact the factor has on the system. In the existing use case points approaches, weights have been assigned based on the experience in their projects [23]. However, we have conducted an online survey to collect the inputs from different practitioners working on SOA and microservices architectures, software architects involved in the migration process and developers working with microservices architecture. We have posted the online questionnaire in multiple social networking platforms including the groups in LinkedIn, Twitter and Facebook etc. The questionnaire included the following questions.

- What is the current role/designation of the participant?
- How much work experience the participant has in SOA and microservices projects?
- Does the participant has real time experience in migration projects?
- How much rating does the participant would like to rate for each of the 21 factors?

The rating of each factor between 0 and 5 for each factor are collected through this survey. Based on the data collected, we have taken the average of ratings and assigned them to all the factors. The weights assigned and ratings of technical and environmental factors are indicated in Table 2 and Table 3.

3-2-3-1-Calculation of Technical Complexity Factor(TCF)

To calculate the TCF, total weight of the 13 factors is calculated which is obtained by multiplying the value assigned to each factor between 0 to 5 and weights assigned to each factor. Calculation of TFactor is given by Eq. (3),

$$TFactor = \sum_{i=1}^{13} TF_i \times W_i \qquad (3)$$

where TF_i is the rating of the technical factor ***i*** and W_i is the weight assigned to corresponding factor. As per the use case points method, the impact of Technical Complexity Factor (TCF) on the proposed service points should vary from a range of 0.6 to 1.3. The formula to calculate TCF is given as below:

TCF = C1 + C2 * TFactor.

Hence, we consider the lowest range value for C1 i.e 0.6 and C2 is calculated as C2 = (1.3-0.6)/50 = 0.014 where 50 is the maximum value of TCF. Therefore, the TCF value for the proposed service point is calculated by the below Eq. (4).

$$TCF = 0.6 + (0.01 \times TFactor) \qquad (4)$$

Table 2: Technical Factors

F_i	*Factors contributing to complexity*	W_i	*Rating*
F_1	Distributed systems	2	5
F_2	Application performance objectives	1	4
F_3	End-user efficiency	1	2
F_4	Complex internal processing	1	2
F_5	Reusability	1	3
F_6	Easy Installation	0.5	1
F_7	Interoperability	0.5	2
F_8	Portability	0.5	1
F_9	Changeability	1	1
F_{10}	Coupling	1.5	5
F_{11}	Modularity	2	4
F_{12}	Statelessness	1	3
F_{13}	Independent deployment	1	4

3-2-3-2-Calculcation of Environmental Factor (EF)

Similarly, the impact of environmental factors in the final service point is evaluated by finding the EF score. To calculate the EF value, the weight of each factor is multiplied with rating assigned to each factor. It is given by the Eq. (5).

$$EFactor = \sum_{i=1}^{8} EF_i \times W_i \qquad (5)$$

The impact of Environmental Factor (EF) is more on the proposed service points method and it varies from a range of 0.0425 to 1.4. The formula to calculate EF is as below:

EF = C1 + C2 * EFactor.

Since, its impact is high, we consider the highest range value for C1 and C2 is calculate as C2 = (1.4-0.0425)/37.5 = 0.03 where 37.5 is the maximum value of EF. The EF for proposed approach is calculated by the below Eq. (6).

$$EF = 1.4 + (-0.03 \times EFactor) \qquad (6)$$

Table 3: Environmental factors

F_i	*Factors contributing to efficiency*	W_i	*Rating*
F_1	Familiar with containers	1.5	3
F_2	Service configurations	1	2
F_3	Analyst capability	0.5	4
F_4	Application experience	0.5	2
F_5	Cloud computing experience	1	2
F_6	Motivation	1	5
F_7	Polyglot	1.5	2
F_8	Stable requirements	1	4

3-2-4-Final service point evaluation

The final Service Points (SP) is calculated by multiplying the unadjusted service point with both technical and environmental factor values. It is given by below Eq. (7).

$$SP = USP \times TCF \times EF \qquad (7)$$

According to Karner [23], the effort required to implement each service point takes 20 hours. Therefore, to estimate the final man-hours, the calculated service point should be multiplied with 20 to get the effort required for migration. Moreover, it is observed that effort required for migrating and designing a microservices application is more compared to designing existing legacy applications [8].

4- Case Study Application

To evaluate and demonstrate the proposed approach, we choose a standard web application that is built based on SOA. In [24], the author has chosen a Vehicle Management System (VMS) application to demonstrate the migration of the legacy application to SOA style. Taking the SOA application as input and applying the microservices extraction approach proposed by Raj, V. *et al.* [20], we have generated the service graph for corresponding microservices based application. The service graph of microservices application is represented in Fig. 3. The service graph is the prototype of a microservices application that has to be built through the migration process. From the service graph represented in Fig. 3, it is clear that there are 12 services in the migrated system. The details of the SOA services, extracted microservices, and the type of services are mentioned in Table 4. As mentioned in Section 3.1, only the migrated service for estimating the effort as few services in SOA based applications can be directly considered as microservices. The calculation of service points according to the proposed approach is presented in the next section.

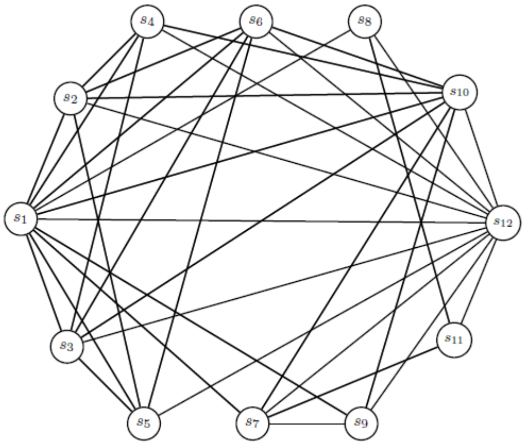


Fig. 3 Service graph representation of microservices based web application

Table 4: Details of extracted microservices from SOA application

SOA Services	*Microservices*	*Notation in service graph*	*Type*
Config Service	Config Service	S_1	Available
Part Service	Part Service	S_2	Available
Product Service	Product Service	S_3	Available
Compare Service	Compare Service	S_4	Available
Incentives & Pricing Service	Incentives Service	S_5	Migrated
	Pricing Service	S_6	Migrated
Dealer & Inventory Service	Dealer Service	S_7	Migrated
	Dealer Locator Service	S_8	Migrated
	Inventory Service	S_9	Migrated
Lead Service	Get-A-Quote Service	S_{10}	Migrated
	Lead Processor Service	S_{11}	Migrated
User Interface Client Service	User Interface Client Service	S_{12}	Available

4-1-Classification of Services

The details of the services along with classification are presented in Table 5. Based on the classification and the weights and ratings of technical and environmental factors, we calculate the service point value used for migration of SOA based application to microservices architecture.

Table 5: List of services along with classification for microservices based application

Service #	*Interacting Services*	*Classification*	*Services considered in estimation*
S_1	2,3,4,5,6,7,9,10, 12	Complex	
S_2	1,4,5,6,10,12	Average	
S_3	1,4,5,6,10,12	Average	
S_4	1,2,3,10,12	Average	
S_5	1,2,3,6,12	Average	✓
S_6	1,2,3,5,10,12	Average	✓
S_7	1,9,10,11,12	Average	✓
S_8	11,12	Simple	✓
S_9	1,7,10,12	Average	✓
S_{10}	1,2,3,4,6,7,9,12	Complex	✓
S_{11}	7,8,12	Simple	✓
S_{12}	1,2,3,4,5,6,7,8,9, 10,11	Complex	

As discussed in Section 2, the services which are classified as migrated services are considered in effort estimation. The other types of services either does not require effort or not suitable in this migration process. Only the services with tick mark will be considered for effort estimation as they are migrated services.

4-2-Calculation of USP

Unadjusted service point value is calculated by multiplying the number of services based on each classification and the weights assigned to each type. From the information from Table 5, there are 2 simple, 4 average and 1 complex services. Therefore, the value of USP is

$$USP = (2 \times 5) + (4 \times 10) + (1 \times 15) = 10 + 40 + 15 = 65.$$

4-3-Considering the Ratings of the Factors Collected Through Online survey

4-3-1-Technical Complexity Factor

First, we need to calculate the TFactor using the information from Table 2. TFactor value is calculated as given below

$$TFactor = \sum_{i=1}^{13} TF_i \times W_i = 46.5$$

Now, we calculate the TCF value.

$$TCF = 0.6 + (0.01 \times TFactor) = 0.6 + (0.01 \times 46.5) = 1.065$$

4-3-2-Environmental Factor

Similarly, we calculate the EFactor using the information from Table 3 and then use this value of EFactor to calculate the EF value.

$$EFactor = \sum_{i=1}^{8} EF_i \times W_i = 23.5$$

Environmental Factor (EF) is calculated by the below equation

$$EF = 1.4 + (-0.03 \times EFactor) = 1.4 + (-0.03 \times 23.5) = 0.695$$

4-3-3-Final service point calculation

The service point is given as the product of USP, TCF, and EF. It is calculated as below.

$$SP = USP \times TCF \times EF = 65 \times 1.065 \times 0.695 = 48.11$$

The total effort required for migrating the SOA based VMS application to microservices is calculated by multiplying the number of services points with 20 hours.

Total estimated effort = 48.11 * 20 ≈ 962 hours.

4-4-Considering the Default Value Suggested by Karner

Karner suggests that if we cannot fill the values for the factors for any reason, we can use the default value as 3 for all the factors [23]. Considering this default value for all factors, we calculated the TCF, EF and service points values.

4-4-1-Technical Complexity Factor

$$TFactor = \sum_{i=1}^{13} TF_i \times W_i = 42.$$

$$TCF = 0.6 + (0.01 \times TFactor) = 0.6 + (0.01 \times 42) = 1.02.$$

4-4-2-Environmental Factor

$$EFactor = \sum_{i=1}^{8} EF_i \times W_i = 24.$$

$$EF = 1.4 + (-0.03 \times EFactor) = 1.4 + (-0.03 \times 23.5) = 0.68.$$

4-4-3-Final service Point Calculation

$$SP = USP \times TCF \times EF = 65 \times 1.02 \times 0.68 = 45.1.$$

Total estimated effort =45.1 * 20 ≈ 902 hrs.

4-5-Observation

By considering the TCF, EF and SP values of both the calculations, the values are very close to each other. Hence, the ratings of factors collected by online survey can be used as reference for estimating the effort of other projects as well.

Table 5: Comparison of values

Ratings	*TCF*	*EF*	*SP*
Collected through online survey	1.065	0.695	48.1
Considering Karner's default value	1.02	0.68	45.1

5-Conclusion

Effort estimation is an important software engineering activity which helps project managers and architects to effectively schedule the project. With the evolution of microservices, companies are migrating existing legacy applications to microservices architecture. In this paper, we propose a new technique which is recasted from the well known use case points technique to estimate the effort required for migration of SOA based applications to microservices architecture. We define a formal model called service graph which is the representation of any service based application. We have revised the technical and environmental factors as the existing factors are not compatible with the microservices architecture. We have conducted online survey to collect the ratings of each of these factors and used in the our effort estimation process. We have demonstrated the new technique through a case study application and calculated the effort required for migration. We have compared the results with effort calculated by considering the default values for factors suggested by Karner.

However, this is the first attempt to estimate the effort required for migration of SOA based applications to microservices and hence, we could not compare it with existing techniques. This approach is applied only on a single case study application and in future we plan to evaluate the proposed technique on applications of different domains and large enterprise applications.

References

[1] Vinay Raj and R. Sadam, “Patterns for Migration of SOA Based Applications to Microservices Architecture,” Journal of Web Engineering, Jul. 2021, doi: 10.13052/jwe1540-9589.2051.

[2] J. Yin, H. Chen, S. Deng, Z. Wu, and C. Pu, “A Dependable ESB Framework for Service Integration,” IEEE Internet Computing, vol. 13, no. 2, pp. 26–34, Mar. 2009, doi: 10.1109/mic.2009.26.

[3] V. Raj and R. Sadam, “Evaluation of SOA-Based Web Services and Microservices Architecture Using Complexity Metrics,” SN Computer Science, vol. 2, no. 5, Jul. 2021, doi: 10.1007/s42979-021-00767-6.

[4] T. Cerny, M. J. Donahoo, and M. Trnka, “Contextual understanding of microservice architecture,” ACM SIGAPP Applied Computing Review, vol. 17, no. 4, pp. 29–45, Jan. 2018, doi: 10.1145/3183628.3183631.

[5] Z. Xiao, I. Wijegunaratne and X. Qiang, "Reflections on SOA and Microservices," 2016 4th International Conference on Enterprise Systems (ES), 2016, pp. 60-67, doi: 10.1109/ES.2016.14.

[6] J. Thones, “Microservices,” IEEE Software, vol. 32, no. 1, pp. 116–116, Jan. 2015, doi: 10.1109/ms.2015.11.

[7] D. Taibi, V. Lenarduzzi, C. Pahl, and A. Janes, “Microservices in agile software development: a workshop-based study into issues, advantages, and disadvantages,” in Proceedings of the XP2017 Scientific Workshops, 2017, pp. 1–5. doi: 10.1145/3120459.3120483

[8] D. Taibi, V. Lenarduzzi, and C. Pahl, “Processes, Motivations, and Issues for Migrating to Microservices Architectures: An Empirical Investigation,” IEEE Cloud Computing, vol. 4, no. 5, pp. 22–32, Sep. 2017, doi: 10.1109/mcc.2017.4250931.

[9] J. Soldani, D. A. Tamburri, and W.-J. Van Den Heuvel, “The pains and gains of microservices: A Systematic grey literature review,” Journal of Systems and Software, vol. 146, pp. 215–232, Dec. 2018, doi: 10.1016/j.jss.2018.09.082.

[10] X. Larrucea, I. Santamaria, R. Colomo-Palacios, and C. Ebert, “Microservices,” IEEE Software, vol. 35, no. 3, pp. 96–100, May 2018, doi: 10.1109/ms.2018.2141030.

[11] P. C. Pendharkar, G. H. Subramanian, and J. A. Rodger, “A probabilistic model for predicting software development effort,” IEEE Transactions on Software Engineering, vol. 31, no. 7, pp. 615–624, Jul. 2005, doi: 10.1109/tse.2005.75.

[12] G. H. Subramanian, P. C. Pendharkar, and M. Wallace, “An empirical study of the effect of complexity, platform, and program type on software development effort of business applications,” Empirical Software Engineering, vol. 11, no. 4, pp. 541–553, Oct. 2006, doi: 10.1007/s10664-006-9023-3.

[13] S. M. Satapathy, S. K. Rath, and B. P. Acharya, “Early stage software effort estimation using random forest technique based on use case points,” IET Software, vol. 10, no. 1, pp. 10–17, Feb. 2016, doi: 10.1049/iet-sen.2014.0122.

[14] S. M. Satapathy, B. P. Acharya, and S. K. Rath, “Early Stage Software Effort Estimation using Random Forest Technique based on Optimized Class Point Approach”, INFOCOMP Journal of Computer Science, vol. 13, no. 2, pp. 22–33, Dec. 2014.

[15] M. R. Braz and S. R. Vergilio, "Software Effort Estimation Based on Use Cases," 30th Annual International Computer Software and Applications Conference (COMPSAC'06), 2006, pp. 221-228, doi: 10.1109/COMPSAC.2006.77.

[16] S. Diev, “Use cases modeling and software estimation,” ACM SIGSOFT Software Engineering Notes, vol. 31, no. 6, p. 1, Nov. 2006, doi: 10.1145/1218776.1218780.

[17] P. Mohagheghi, B. Anda and R. Conradi, "Effort estimation of use cases for incremental large-scale software development," Proceedings. 27th International Conference on Software Engineering, 2005. ICSE 2005., 2005, pp. 303-311, doi: 10.1109/ICSE.2005.1553573.

[18] G. Canfora, A. R. Fasolino, G. Frattolillo, and P. Tramontana, “A wrapping approach for migrating legacy system interactive functionalities to Service Oriented Architectures,” Journal of Systems and Software, vol. 81, no. 4, pp. 463–480, Apr. 2008, doi: 10.1016/j.jss.2007.06.006.

[19] Z. A. Siddiqui and K. Tyagi, “A critical review on effort estimation techniques for service-oriented-architecture-based applications,” International Journal of Computers and Applications, vol. 38, no. 4, pp. 207–216, Oct. 2016, doi: 10.1080/1206212x.2016.1237132.

[20] V. Raj and R. Sadam, “A Framework for Migration of SOA based Applications to Microservices Architecture,” Journal of Computer Science and Technology, vol. 21, no. 2, p. e18, Oct. 2021, doi: 10.24215/16666038.21.e18.

[21] E. A. Farrag, R. Moawad, and I. F. Imam, “An Approach for Effort Estimation of Service Oriented Architecture (SOA) Projects,” Journal of Software, vol. 11, no. 1, pp. 44–63, 2016, doi: 10.17706/jsw.11.1.44-63.

[22] A. Yanchuk, A. Ivanyukovich, and M. Marchese, “Towards a Mathematical Foundation for Service-Oriented Applications Design,” Journal of Software, vol. 1, no. 1, Jul. 2006, doi: 10.4304/jsw.1.1.32-39.

[23] G. Karner, “Resource estimation for objectory projects,” Objective Systems SF AB, vol. 17, pp. 1–9, 1993.

[24] P. Bhallamudi, S. Tilley and A. Sinha, "Migrating a Web-based application to a service-based system - an experience report," 2009 11th IEEE International Symposium on Web Systems Evolution, 2009, pp. 71-74, doi: 10.1109/WSE.2009.5630392.

[25] F.S. Aliee, S. Oviesi. A way to improve Adaptive Maintenance in Enterprise Architecture. Journal of Information Systems and Telecommunication;8(1):1-4.2020;doi: 10.7508/jist.2020.01.001

[26] V. Raj and S. Ravichandra, "Microservices: A perfect SOA based solution for Enterprise Applications compared to Web Services," 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2018, pp. 1531-1536, doi: 10.1109/RTEICT42901.2018.9012140.

[27] V. Raj and R. Sadam, “Performance and complexity comparison of service oriented architecture and microservices architecture,” International Journal of Communication Networks and Distributed Systems, vol. 27, no. 1, p. 100, 2021, doi: 10.1504/ijcnds.2021.116463.

Journal of Information Systems and Telecommunication
Vol.10, No.2, April-June 2022, 89-101

http://jist.acecr.org
ISSN 2322-1437 / ESSN:2345-2773

Hierarchical Weighted Framework for Emotional Distress Detection using Personalized Affective Cues

Nagesh N Jadhay[1*], Rekha Sugandhi[2]

[1].Department of Computer science and Engineering, MIT School of Engineering, MIT ADT University, India
[2].Department of Information Technology, MIT School of Engineering, MIT ADT University, India

Received: 09 Jul 2021 / Revised: 04 Oct 2021 / Accepted: 08 Nov 2021
DOI:

Abstract

Emotional distress detection has become a hot topic of research in recent years due to concerns related to mental health and complex nature distress identification. One of the challenging tasks is to use non-invasive technology to understand and detect emotional distress in humans. Personalized affective cues provide a non-invasive approach considering visual, vocal, and verbal cues to recognize the affective state. In this paper, we are proposing a multimodal hierarchical weighted framework to recognize emotional distress. We are utilizing negative emotions to detect the unapparent behavior of the person. To capture facial cues, we have employed hybrid models consisting of a transfer learned residual network and CNN models. Extracted facial cue features are processed and fused at decision using a weighted approach. For audio cues, we employed two different models exploiting the LSTM and CNN capabilities fusing the results at the decision level. For textual cues, we used a BERT transformer to learn extracted features. We have proposed a novel decision level adaptive hierarchical weighted algorithm to fuse the results of the different modalities. The proposed algorithm has been used to detect the emotional distress of a person. Hence, we have proposed a novel algorithm for the detection of emotional distress based on visual, verbal, and vocal cues. Experiments on multiple datasets like FER2013, JAFFE, CK+, RAVDESS, TESS, ISEAR, Emotion Stimulus dataset, and Daily-Dialog dataset demonstrates the effectiveness and usability of the proposed architecture. Experiments on the enterface'05 dataset for distress detection has demonstrated significant results.

Keywords: Convolution Neural Network; Long Short-Term Memory; Transformers; Hierarchical Fusion; Distress Detection.

1- Introduction

In the current scenario of a global pandemic, every individual is fighting his or her own battles on financial, social, and medical fronts. The current situation is making humans undergo emotional fluctuations, resulting in challenges in mental health. Paul Ekman proposed seven basic emotions like happy, sad, fear, angry, disgust, surprise and neutral [1]. Every person undergoes various emotions based on the context he or she is in. It is natural for a human to exhibit different emotions, however, the prevalent appearance of some emotions like sadness, fear and anger can be alarming. For example, consider a small video from Youtube of 10 seconds with 40 frames and 10 frames per second frame rate, the total number of emotional changes that happened are around 40. The term 'distress' is related to the affective state that arises in vicious or furious situations [2]. The National Comprehensive Cancer Network has defined distress. They quote 'Distress is a multifactorial unpleasant emotional experience of a psychological (cognitive, behavioral, emotional), social, and/or spiritual nature. Distress encompasses a range of common feelings of vulnerability, sadness, and fears that can cause depression, anxiety, panic, social isolation, and existential and spiritual crisis' [3]. Unanticipated external events are not only the cause of distress but also internal states like thoughts, chronic behaviors, and feelings of the person. Depression and anxiety disorders follow the prevalent distressed condition.

The affective computing community is working closely with psychologists in the identification, prevention, and treatment of mental disorders. Recent studies signify the use of complex multi-modal systems preferred over single modal architectures. Humans express and communicate their emotions in a multimodal way. Affective cues such as visual, auditory, and textual are exploited parallelly and cognitively to extract affective and profound information communicated during the interaction. This information can be used to identify emotional distress if analyzed and used

✉ **Nagesh N Jadhay**
nagesh.jadhav@mituniversity.edu.in

effectively. Mehrabian and Ferris, in their research, revealed the contribution of audio, visual and textual cues during the communication. According to the authors, 55% of communication is visual, 38% is vocal and 7% is verbal [4]. In recent years, people are inclined towards social media, which adds to a large amount of video, audio, and text data generation. Social media platforms are also used to express opinions about products, services, or people. Information can be analyzed to extract required details to identify the emotional states of the person. The multi-modal frameworks are still a challenge in computer science due to two main reasons. 1) It is difficult to extract useful features from the audio, video, and text data. 2) The different dimensions of each modality make it difficult to fuse the feature and process it.

In this paper, we are proposing a hierarchical weighted framework for the detection of emotional distress based on audio, video, and text modalities. Most of the literature focuses on multi-modal emotion recognition using video and audio cues. We have also considered textual cues to understand the affective state of the person. Considering the complexities and challenges in facial emotion detection we have employed a hybrid approach on multiple face emotion datasets. We have used two native convolutional neural networks (CNN) approaches and one transfer learning approach to achieve better results. The purpose behind using multiple datasets is to have a wide variety of facial expressions from people across the globe. We have also developed our dataset of face emotions expressions, where all the images are taken in wild with an available resource like a smartphone. Over 43 students from the computer science and engineering department volunteered for this activity. For audio affective cues, we used both convolutional neural networks and Long Short-Term Memory (LSTM) architectures. LSTM performed reasonably well in comparison to CNN. The textual cues are also very challenging to handle. Wrong interpretation of context will lead to the wrong semantic of the sentence. Initially, we worked with bi-directional LSTM, however, we found Bidirectional Encoder Representations from Transformers (BERT) are very effective and precise for textual cue processing. Fusing the results from the different modalities is a challenging problem. To fuse the results, we have proposed a novel hierarchical weighted framework. To detect the emotional distress of a person we have also proposed and implemented the algorithm based on the negative emotions demonstrated by the person.

2- Related Work

Emotional distress has become one of the common mental illnesses if goes unattended may lead to anxiety, depression moreover into suicidal tendencies. Prolonged stress also results in sleeplessness, mood swings and lack of attention. Emotional distress is directly related to mental disorders. The clinical approaches in diagnosing the distress include manual intervention and questionaries to answer. Automating this process with optimized deep learning approaches would be groundbreaking and it will help medical and psychological practitioners significantly.

In past, affective state recognition including facial cues and audio cues been studied by many researchers. Most of the work focuses on the utilization of audio and video cues ignoring textual affective cues. Thomas et al. [5] have used audio and visual information to understand the affects. The research focuses on identifying valence and arousal for the affects. Dataset used is the MediaEval 2015 dataset. Both audio and video representations are fused and trained using an SVM classifier. Each of the features is also trained independently and used an ensemble approach to fuse the results. The combining video and audio features have demonstrated good results compared to handling a single modality. Affective internet of things is an active area of research that can detect the affective state of the human. Wearable systems which are the combination of recent market technology and smart sensors are an important part of affective computing. Miranda et al. have used Galvin skin resistance and blood volume pulse as modalities to detect fear emotion, which falls in the negative quadrant of the valence and arousal. Using wearable computing for affective state recognition could be an expensive affair when it is considered for real-time use and applications [6]. The authors [7] have proposed a multi-modal fusion scheme of audio-video features to detect the depression of the person. The extracted audio and video features from the stream of data, initially processed using principal component analysis to reduce the redundancy, which is further provided as input to the epsilon support vector regression model for the prediction. The predicted values are finally combined with the local linear regression model to predict the result. For visual cues divergence curl shear descriptors, space temporal interesting points and head pose features are used as the features to detect depression. For the audio motion history histogram, a bag of words and vector of locally aggregated descriptors are used to examine the audio of the person. The proposed method uses the 3DCLS hybrid model which is the combination of 3D CNN and convolutional LSTM. The text modality is processed using the CNN-RNN hybrid model. CNN is used to extract features from the text while RNN is used to predict the emotion. For the visual data, C3D is used to extract Spatio-temporal features and CNN LSTM predicts the feature sequences. Features from the audio data are extracted using the OpenSmile tool and processed using SVM [8]. Zeinab and Saeed proposed the MoBEL model [9], which is a combination of expert neural networks and brain-inspired learning algorithms. It works in two stages. In the first stage, CNN and RNN are applied to extract high-level features while in the second stage model is trained to learn audio-video features. Luu-Ngoc Do et al. in their work used a hybrid CNN RNN model to detect the facial emotions of the person. The JAFFE and MMI are the datasets used for the experimentation. The modality considered is limited to only visual cues [10].

Jain et al. used a hybrid CNN RNN model to detect the facial emotions of the person. The JAFFE and MMI are the datasets used for the experimentation. The modality considered is limited to only visual cues [11]. Kaya et al. have proposed a multimodal approach where the input is processed to extract faces from the video. The extracted faces are trained using a pre-trained VGG16 model. The audio cue is processed separately, and results are fused at the decision level. For classification model learning, kernel extreme learning machine and partial least square regression are used. Finally, the results are combined using a weighted fusion of scores [12]. Zhang et al. discuss the multi-modal approach for speech emotion recognition. The proposed method creates three audio inputs for the CNN model. The First 1D CNN model uses raw waveform, 2D CNN uses MFCC feature while 3D CNN exploits temporal-spatial features of the input. The score level fusion is performed to attain the final prediction [13]. Yan et al. have defined a multi cue fusion emotion recognition framework based on the audio signal, facial texture, and facial landmarks. To capture change in facial texture the cascaded convolutional neural network and Bi-directional RNN is used. Audio features are extracted using CNN and stored in a matrix for further processing. The faces are trained using a pre-trained VGG16 network. The RNN model is employed on extracted facial texture feature to recognize dynamic differences in facial feature sequences. For audio, OpenSmile is used to extract the features, further CNN is used for classification. Both feature level fusion and decision levels strategies are exploited to achieve results [14]. Bendjoudi et al. proposed architecture for the visual cues. An input image is processed by a scene detector module. Body module is used to process cropped images. VGG16 pre-trained network and Xception network is used for scene detector and body detector module, respectively. Authors have proposed the multi-label focal loss function to calculate the loss on the Emotic dataset [15]. Hao et al. proposed an emotion recognition method based on multi-task and ensemble learning using audio-video features. Both the deep features and manual features of audio-video cues are extracted and presented to different algorithms for processing. The proposed architecture consists of four models, CNN for mel spectrogram, SVM from Interspeech2010 features, CNN for facial emotion recognition and SVM for LBP features. A blending algorithm is used to fuse the results from different classifiers [16]. Tzirakis et al. have considered visual and auditory cues for emotional recognition. CNN is used to extract features from speech while pre-trained ResNet50 is used to extract visual features. The RECOLA dataset has been used to test the developed model. The output from both modalities are combined in a feature vector and passed to LSTM for the prediction of the result [17].

Majumder et al. have worked on three modalities like audio, video, and text. In the initial phase, features are extracted individually from each of the modalities. The proposed hierarchical approach starts with a bimodal fusion of modalities followed by trimodal fusion. The experiments were performed on the IEMOCAP and CMU-MOSI databases [18]. The proposed method in [19] considers three modalities, audio, video and text for data analysis. The experiments were performed on e'NTERFACE database. Each modality is trained individually on a different dataset. For visual, auditory and text-based cues, CK+ and ISEAR databases are used respectively. Yaxiong Ma et al. proposed a deep fusion method for audio and visual cues. To remove cross pollution in audio data and redundancy in visual cues, cross-modal noise modelling has been used. Audio feature extraction is done using 2D CNN while video feature extraction is done using 3D CNN. Finally, deep belief networks are employed for the nonlinear fusion of extracted features [20]. Guo and the team worked on video and audio data for video content analysis. VGG16 pre-trained is used to extract the video features and audio features are extracted using the OpenSMILE tool. Following end-to-end training, two subtasks are introduced i.e., classification and regression. The classification task focuses on classifying fear-induced videos and the regression task predicted the arousal and valence values of the user [21]. Poria et al. used audio, video, and text data to perform sentiment analysis of the video. Both feature level fusion and decision level fusion are utilized to merge different modalities [22]. Noroozi et al. proposed architecture to work on the audio and video data. For audio data features like MFCC, prosodic and filter bank energies are extracted and for visual cues, CNN is applied to extract the required features. Finally, the confidence outputs of multiple classifiers from the different modalities are fused to fetch results [23]. In [24] Audiovisual cues are considered to learn affective features using a combination CNN and 3D CNN model. The extracted audio-video features are fused using the DBN network. Li and Liu [25] have developed 1D CNN network and multi-layer perceptron network to detect stress using physiological signals. The input has been classified into baseline, stressed and amused states. Bobade and Vani have worked on multimodal physiological signals for the detection of stress using machine learning algorithms. Results in the paper demonstrated deep neural networks performs better compared to conventional machine learning algorithms [26]. Zhang et al. have discussed video based stress detection. The authors have used CNN for the implementation. The results of facial expression and facial units are combined using weighted integration of local and global attention [27].

The literature survey has identified the following research gaps, which are addressed in this paper.

- Most of the multi-modal approaches focus on physiological signals.
- Results of training and testing are demonstrated on the same dataset. Cross data result validation is missing in many approaches
- The early feature fusion approach is used by most researchers which raises the requirement of high computational resources for model training.
- We found limited literature related to stress detection using deep learning and late fusion approaches.

3- Proposed Work

The overview of the research methodology is shown in Fig. 1. Where audio-video input has been given to the system for processing. In the initial step video frames, audio, and text data has been separated from the input and assigned to subsequent block for further feature extraction and processing. The working of each block has been discussed in detail in the following sections.

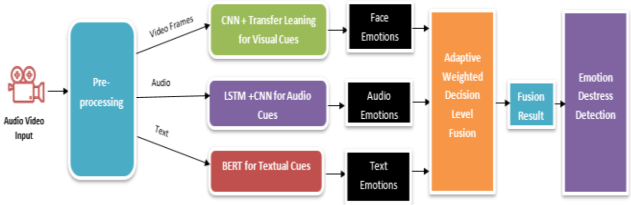


Fig. 1 Proposed Work

3-1- Visual Cues

Most of the literature has identified that the facial cues are very challenging to work due to unavailability of the data, variation facial structures, ethnicity of the person, wrong perception of emotion by annotators of dataset etc. Many datasets also have a biased distribution of emotions in the dataset. To overcome these issues, we have worked with three different datasets like Face Emotion Recognition 2013 (FER 2013), Japanese Female Facial Expressions (JAFFE) and Cohn Kanade plus (CK+) datasets [28][29][30]. A detailed overview of facial cues processing and prediction is shown in Fig. 2.

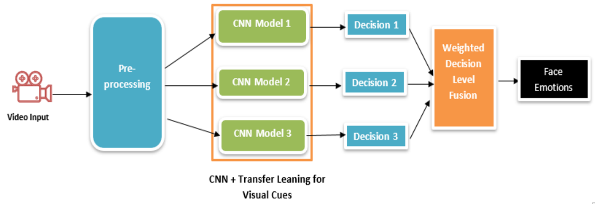


Fig. 2 Visual cue processing for facial emotion recognition

3-1-1- Preprocessing

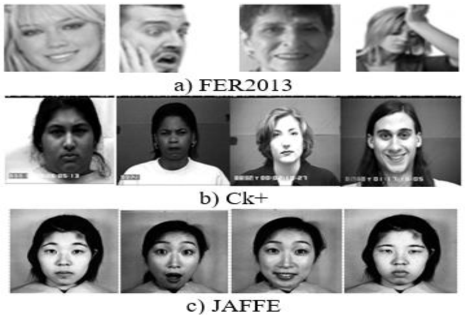
The video data received is in different formats and dimensions. The preprocessing is performed on the video to extract the faces from the video frames. This is done using two steps:

1. Face Detection: Many face detection algorithms are available to detect and extract faces from video frames. We have worked with two approaches classical Voila-Jones algorithm and multitask cascaded convolutional networks. Both algorithms worked well for the input provided. Considering the amount of time and memory required to process the input, we decided to use a lightweight Voila-Jones algorithm.

2. Normalization: The detected faces are resized and normalized as per the requirement of CNN models. Each detected face has been resized to (197,197,3), (128,128,3), and (48,48,3) respectively.

3-1-2- Datasets

For face emotion recognition we have used three datasets, FER2013, JAFFE and CK+. The FER2013 is an extensive dataset with 35887 grey images of dimension (48*48*1) depicting 7 basic emotions. The distribution of classes is given as Angry:4593, Disgust:547, Fear:5121, Happy:8989, Sad:6077, Surprise:4002, and Neutral:6198 images. We can see from the distribution that the dataset has a small number of images depicting 'disgust' emotion. The JAFFE dataset has 213 images of Japanese females portraying 7 basic images. All the images are of dimension 256*256*1. The third dataset we have used for model training is the CK+ dataset. It is a very popular dataset consisting of 593 image sequences. Apart from basic emotion, it also has images representing 'contempt' emotion. The sample pictures from the database are shown in Fig. 3.



a) FER2013

b) Ck+

c) JAFFE

Fig. 3 Facial Cue Datasets

3-1-3- CNN Model I

For the first CNN model, we have adopted already trained ResNet50 [31] and VGGFace model, instead of developing the model from the scratch. ResNet50 is one of the popular models used in computer vision. It consists of 48 convolution layers, one max pooling and one average pooling layer. The ResNet50 was developed to overcome the problem of vanishing gradient and results in saturation as the model goes deep in architecture. ResNet50 was trained on the Imagenet database and can classify 1000 objects. The key part of ResNet50 is identify block which takes residual connections. The Working of identify block is shown in Fig. 4.

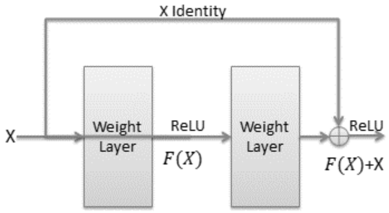


Fig. 4 Residual Identity Block

The Resnet50 has achieved great success in recognizing required features from the images automatically. VGGFace model has performed greatly in recognizing the faces. Therefore, the already trained ResNet50 model along with the weight of VGGFace can be easily transferred to learn the feature of the images from the FER2013 dataset. However, the images in the FER2013 dataset are of size 48*48*1. All these images are required to resize to the dimension that suits ResNet50 and VGGFace models. Each image has been resized to dimension (197,197,3). To avoid overfitting, we employed image augmentation to generate runtime tensor image data. The top layers of the model are fully trained by freezing convolutional layers. After top layer training, the convolutional layers of ResNet50 are fine-tuned to achieve high accuracy for both validation and test set.

3-1-4- CNN Model II

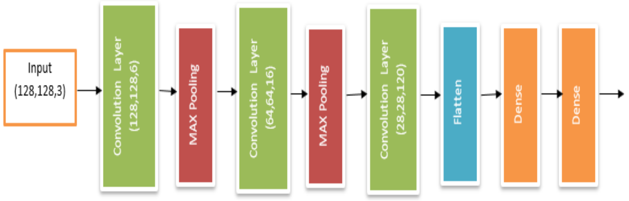

Input (128,128,3)
Convolution Layer (128,128,6)
MAX Pooling
Convolution Layer (64,64,16)
MAX Pooling
Convolution Layer (28,28,120)
Flatten
Dense
Dense

Fig. 5 Architecture for JAFFE dataset

The second model is designed for the JAFFE database. The FER2013 dataset has very limited images displaying 'disgust' emotion. To avoid misclassification of disgust emotion we have incorporated the JAFFE dataset. The CNN model is developed and fine-tuned to work on the JAFFE dataset. The basic architecture is shown in Fig. 5. The input of dimension (128,128,3) has passed to the CNN architecture. The model consists of three convolutional layers, max-pooling layer is interspersed between the convolutional layers to downsample the feature map. The flatten is layer followed by two dense layers and softmax activation is used as the output unit. The softmax activation is denoted by the following expression,

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{i=1}^{N} e^{z_i}} \quad (1)$$

$$z_i = weight \times input^T$$

3-1-5- CNN Model III

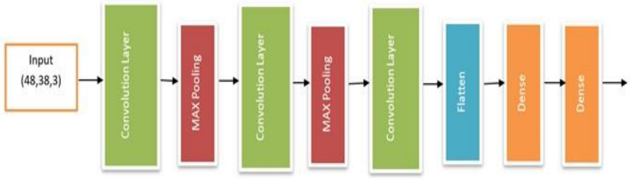


Fig. 6: Architecture for CK+ dataset

To improve the overall accuracy of facial emotion recognition, we have considered another CNN model on the CK+ dataset. All images are resized to (48,48,3) dimensions. To split the data, k-fold validation has been used with a value for a number of splits is equal to 5. The model has shown significant accuracy in the training and testing phase. The model architecture with dimensions is displayed in Fig. 6. All the models produce their own decision upon processing the various dimension inputs. All decisions are fused using a weighted approach.

3-1-6- Visual Cue Emotion Recognition

Emotions = {Anger, Disgust, Fear, Happy, Sad, Surprise, Neutral}
TA_video = {w1, w2, w3}
Where TA_video is normalized test accuracy of video modality for CNN model 1, model 2 and model 3 respectively. The weight normalization is given as,

$$w_i = \frac{M_{acc}}{Acc_{Total}} \tag{2}$$

Where M_acc is the test case accuracy of the model.
Steps for the emotion recognition from video are given below
Step 1: Video input
Step 2: Pre-process the input video to extract the faces from video frames and resize the frames as per required dimensions.
Step 3: Classification – Apply all models concurrently to classify the video frame in one of the classes from Emotions.
Step 4: Apply the voting approach to select the final facial emotion
Step 5: In the case of 3 indifferent decisions, apply weights w1, w2, and w3 respectively to each model's decision.
Step 6: Make final prediction using following equations,

$$face_{emotion} = argmax\left\{\sum_{1}^{L} argmax\left(w1 \times Decision_1, w2 \times Decision_2, w3 \times Decision_3 \right)\right\} \tag{3}$$

Where $Decision_1$, $Decision_2$, *and* $Decision_3$ are the probability values associated with each decision, L is the length of the video. Based on the emotion of each frame detected, higher count emotion is selected as the final $face_{emotion}$ of the video, ignoring neutral emotion.

3-2- Audio Cues

Many times, a person fakes their facial expressions to hide the actual emotions he is undergoing. So, relying on visual cues would add inconsistencies in the recognition of emotional distress. Audio cues, when combined with visual cues provide significant results. Different audio features describe different emotions. For example, anger emotion can be defined using pitch frequency, rapid speech rate and high energy. Similarly, sadness can be described as low energy and pitch frequency. Literature suggests Mel Frequency Cepstral Coefficients (MFCC), spectral energy distribution, the intensity of speech, pitch, Zero Crossing Density (ZCD) are the important and widely used features to recognize emotions from audio signals [32]. Mel Frequency Cepstral Coefficient: Mel scale is related to the perceived frequency or pitch of actual measured frequency. The Mel scale is represented using the following formula,

$$M(f) = 1125 \times \ln\left(1 + \frac{f}{700}\right) \tag{4}$$

Given the above equation, Mel frequency is represented as,

$$M^{-1}(m) = 700 \times \left(exp\left(\frac{m}{1125}\right) - 1\right) \tag{5}$$

The Cepstral Coefficients (CC) can be exploited to split the original signal from the filter. Spectral details of a signal can be extracted by truncating the signal at different frequencies. To calculate Cepstrum, Discrete Fourier Transform (DFT) of the log magnitude of the DFT of the signal is calculated. The model proposed by Davis et al. [33] to calculate MFCC is given in equation (6),

$$MFCC_i = \sum_{\theta}^{N} \cos\left(i(\theta - 1)\frac{\pi}{N}\right),\ i = 1, 2, 3, \ldots N \tag{6}$$

Where M is cepstral coefficients, Theta is long energy output and N is a number of triangular bandpass filters.
For audio cues, we developed two models, the first model exploiting the capabilities of CNN and the second model using the LSTM approach. Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [34] and Toronto emotional speech set (TESS) [35] datasets of emotional audio speech are used. The proposed architecture for audio cue recognition is shown in Fig. 7.

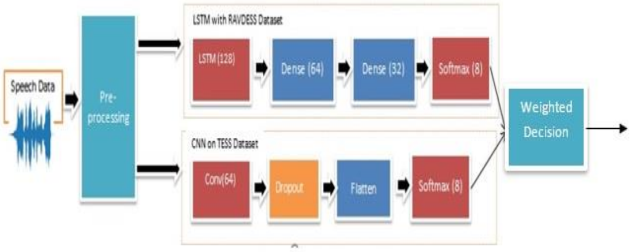


Fig. 7 Architecture for Audio cues emotion recognition

We employed the CNN model on the TESS dataset and the LSTM model on RAVDESS dataset. We used 40 MFCC features in both models for preprocessing. Audio data has been extracted from video and preprocessed to fetch MFCC features from the input. The results of both approaches are discussed in the experimentation and results section. However, the LSTM model outperformed CNN model in terms of accuracy and generalization. We fused results of both the models using a weighted approach where higher weight is assigned to the LSTM model based on the test accuracy achieved. See equation (7).

$$Audio_{decision} = argmax\left[w1_{cnn} \times P(D_1), w2_{lstm} \times P(D_2)\right] \tag{7}$$

3-2-1- Datasets

Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS), the dataset contains 7356 files for 24 professional actors, 12 male and 12 female. The speech data depicts happy, calm, sad, angry, fear, surprise, and disgust emotions. The intensity of emotions portrayed is both normal and high. For the experimentation, we have used speech audio files (1440 files). Toronto emotional speech set (TESS), the dataset contains speech data, spoken by two actresses over 200 target words. It also portrays seven basic emotions.

3-3- Textual Cues

Recognizing affective states or emotions from the text is a challenging task. The reasons for the same can be listed as, word ambiguities, the complexity of meanings, writing style, different languages, different cultures and many more. Emotion detection from a text can be done using a rule-based approach, machine learning approach or hybrid approach. While recognizing emotion from the text, it is also necessary to understand the semantics and context of a sentence. We used a bi-directional LSTM and BERT transformer for the experimentation on text datasets. However, the BERT transformer performed significantly well in comparison with bi-directional LSTM [36]. We used International Survey on Emotion Antecedents and Reactions (ISEAR) dataset [37], Emotion Stimulus dataset [38] and DailyDialog dataset [39]. For the textual cues affect detection we considered only emotions, neutral, happy, sad, angry and fear.

3-4- Decision Level Hierarchical Weighted Fusion and Distress Detection

3-4-1- Decision Level Hierarchical Weighted Fusion

One of the key challenges in the multi-modal approach is to fuse the different modalities to generate a result. The majority of the literature suggests three approaches i.e., early fusion, late fusion, and hybrid fusion. A most common solution to fuse different modalities is to have early or feature level fusion. Early fusion suffers from several drawbacks. If we have a large number of features fusing them will lead to low accuracy if the training dataset is small. Format incompatibility and temporal features mapping would be challenging for early fusion. Lastly, the requirement of huge computational resources to manage high dimensional data is another significant challenge [40] [41]. In this paper, we have utilized decision level hierarchical weighted fusion. We have observed each modality has a different way of learning parameters. This results in variation in the accuracy values of each modality. So, to have consistency in decision making, we have considered the test accuracy of the datasets as a weighted parameter for further calculation. We created a hierarchy of decisions before making a final decision. The rule for the calculation of visual cue prediction probability is given in equation (8),

$$
\begin{aligned}
&P(F_E) = argmax\left[w_1 \times P(model_{f1}), w2 \times P(model_{f2}), w_3 \times P(model_{f3})\right]\\
&P(A_E) = argmax\left[w_1 \times P(model_{a1}),\ w_2 \times P(model_{a2})\right]\\
&P(T_E) = argmax(\sigma(z_i))
\end{aligned}
\tag{8}
$$

Where wi is normalized as per the formula given in equation (2). P(FE), P(AE) and P(TE) are the probabilities of face, audio, and text emotions, respectively. The result of all fused modalities is given as,

$$
\begin{aligned}
&Affective_{state} = argmax\left\{W_1 \times \left[\frac{P(F_E)}{S_A}\right], W_2 \times \left[\frac{P(A_E)}{S_A}\right], W3 \times \left[\frac{P(T_E)}{S_A}\right]\right\}\\
&where,\ S_A = P(F_E) + P(A_E) + P(T_E)
\end{aligned}
\tag{9}
$$

For the final decision, we have considered static weight values (W1, W2, W3) assigned to visual, vocal, and verbal cues, respectively. We have followed the Mehrabian and Ferris approach for the contribution of each cue in the communication. We tweaked the standard values mentioned and derived new weight values. For visual cues, we have considered 0.40, for audio 0.35 and text 0.2.

3-4-2- Emotional Distress Detection

Emotions are a vital part of human life. Every human feels all basic seven emotions, those may be comfortable or uncomfortable. Distress is different from emotions. We undergo different emotions regularly and we can observe many fluctuations in the emotion. The negative emotions like sadness, fear, anger, and disgust if observed consistently, it is an indication of a threatening situation. If left unattended, this emotional combust may lead to severe mental health problems. Here we are proposing the automated approach to detect emotional distress. Our system will be observing and analyzing the affective states of the person. More specifically, negative emotions like fear, sadness and anger need to be observed for their prevalent occurrences. Our proposed system will alert the person about his frequent emotional changes. The detailed flowchart is shown in Fig. 8.

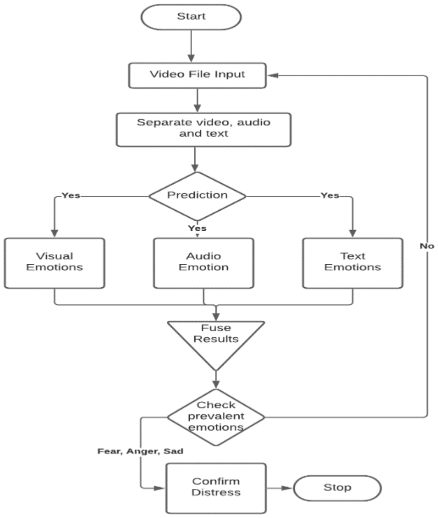


Fig. 8 Emotional Distress Detection

4- Experiments and Results

We have conducted experiments on various datasets depending on the various affective cues. Table 1 below shows different datasets we have used.

Table 1: Affective cues and datasets

Sr. No.	Affective cue/ Modality	Datasets
	Visual cues	FER2013 CK + JAFFE
	Vocal Cues / Audio data	RAVDESS TESS
	Verbal cues / Text data	ISEAR Daily Dialog Emotional Stimulus

4-1- General Settings

Given a video clip as input, data is extracted from it. Audio is extracted from video and audio is converted into text format for further processing. An entire experiment has been run into a Google collaboratory environment with 12 GB of RAM and GPU support. We have limited video length to 40 frames with frames per second rate varies from 15 fps to 25 fps.

4-2- Experiments on Visual Cues

4-2-1- Experiments with Fer2013 Dataset

FER2013 dataset is very challenging to work on because of biased distribution of samples, mislabeled emotions etc. Also, most of the models we developed, were overfitting for the FER2013 dataset. We have used transfer learning using the pre-trained ResNet50 model and weights of the VGGFace model to detect faces and recognize emotions. Every input image has been normalized to the mean of the dataset. Input has been resized to the dimension of (197,197,3). For better accuracy, we applied image augmentation with (rotation range, shear range, zoom range, horizontal flip) parameters. We have used Adam optimizer with a learning rate of 0.0001 and epsilon value with (1*e-08). With the fine-tuned model we have achieved 71.25% test accuracy. The confusion matrix and classification report are shown in Fig. 9 and Table 2, respectively.

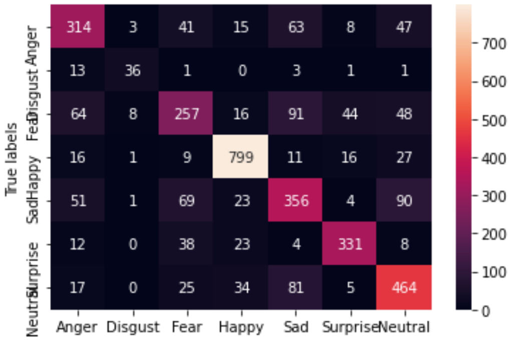


Fig. 9 CNN model 1 confusion matrix

Table 2: CNN Model 1 Classification Report

	precision	recall	f1-score	support
Anger	0.64	0.64	0.64	491
Disgust	0.73	0.65	0.69	55
Fear	0.58	0.49	0.53	528
Happy	0.88	0.91	0.89	879
Sad	0.58	0.6	0.59	594
Surprise	0.81	0.8	0.8	416
Neutral	0.68	0.74	0.71	626
Accuracy			0.71	3589
Macro Avg	0.7	0.69	0.69	3589
Weighted Avg	0.71	0.71	0.71	3589

4-2-2- Experiments with JAFEE Dataset

Anger, fear and sadness are the important emotions considered for distress detection. From the results of FER2013 dataset experiments, it is observed that accuracy for these negative emotions varies between 53% to 64%. To improve the overall performance of visual cues recognition, we have considered the JAFFE dataset. JAFFE is a relatively small dataset consisting of 213 images. We have developed CNN models with dropout layers interspersed between convolution layers to avoid overfitting of the model. For this model, we have used the RMSProp optimizer and cross-entropy loss function. Model is trained for 30 epochs with a batch size equal to 8. The validation accuracy achieved on the JAFFE dataset is 84.37%. The confusion matrix for the JAFFE dataset is shown in Fig. 10.

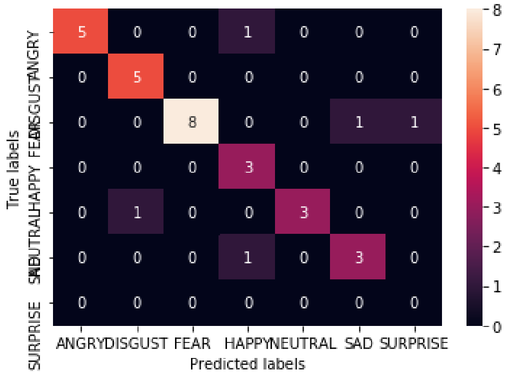


Fig. 10 CNN model 2 confusion matrix

The precision, recall and f1 score achieved for model 2 is 91%, 84% and 86% respectively, shown in Table 3.

Table 3: CNN Model 2 Classification Report

	precision	recall	f1-score	support
Anger	1.00	0.83	0.91	6
Disgust	0.83	1.00	0.91	5
Fear	1.00	0.80	0.89	10
Happy	0.60	1.00	0.75	3
Neutral	1.00	0.75	0.86	4
Sad	0.75	0.75	0.75	4
Surprise	0.00	0.00	0.00	0
Accuracy			0.84	32
Macro Avg	0.74	0.73	0.72	32
Weighted Avg	0.91	0.84	0.86	32

4-2-3- Experiments with CK+ Dataset

The CNN model 3 we have developed utilizes the CK+ dataset for experimentation. While working with other datasets, we observed that the perception and demonstration of emotion vary from person to person. So, to avoid misclassification errors, we have used the CK+ dataset to make our architecture enrich with a dataset. All the images from the dataset are resized to dimension (48,48,3). We have used an RMSProp optimizer with default learning rate and cross-entropy as loss function. We have achieved 73.09% test accuracy. The confusion matrix is shown in Fig. 11. It is notable from Table 4 that, training and validation sets are unbalanced, which resulted in poor classification for some classes. From Table 4 we can observe that 'Contempt' emotion is hard to classify. It is either classified as 'Anger' or 'Disgust'. ‘Contempt’ emotion is part of only CK+ dataset, so for experimentation purposes, we have ignored the 'Contempt' emotion. Happy emotion is having good prediction accuracy than other emotions.

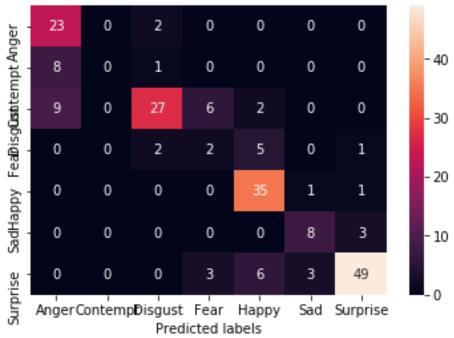


Fig. 11 CNN model 3 confusion matrix

Table 4: CNN Model 3 Classification Report

	precision	recall	f1-score	support
Anger	0.57	0.92	0.71	25
Contempt	0.00	0.00	0.00	9
Disgust	0.84	0.61	0.71	44
Fear	0.18	0.20	0.19	10
Happy	0.73	0.95	0.82	37
Sad	0.67	0.73	0.70	11
Surprise	0.91	0.80	0.85	61
Accuracy			0.73	197
Macro Avg	0.56	0.6	0.57	197
Weighted Avg	0.73	0.73	0.72	197

To validate the visual cues architecture, we have created our dataset named ‘Emo-CSE Database’. The dataset consists of the facial expressions of 43 plus student participants from the Computer Science and Engineering department. For the experiments, participants were asked to click their picture in a natural environment and annotate it with respective emotions. Most of the pictures were taken using a smartphone camera with no standard pre-settings. The samples from the dataset are shown in Fig. 12.

Fig. 12 Emo-CSE database

Overall accuracy achieved on the validation dataset is 76.24 %. Table 5. shows the performances of models for visual cues.

Table 5: Visual Cue Model Performances

Method	Accuracy (%)
CNN Model 1	71.25
CNN Model 2	84.37
CNN Model 3	73.09
Weighted Fusion	77.24

4-3- Experiments on Audio Cues

4-3-1- Experiments with RAVDESS Dataset

RAVDESS dataset contains audio-video files of song and speech data. We have considered only speech data for the experimentation. From the input speech data, we extracted 40 MFCC features of audio and used them to train the LSTM network. We trained the network for 100 epochs with Adam optimizer and cross-entropy as loss function. LSTM model achieves 79.51% test accuracy. The confusion matrix is shown in Fig. 13. Table 6 depicts the classification report for RAVDESS test set.

4-3-2- Experiments with TESS Dataset

For the TESS dataset, we have used the CNN model to process the audio files. 40 MFCC features are extracted from the input audio file with shape (40,1) and fed into a 2D convolution layer. We have a sparse categorical cross entry loss function and Adam optimizer. The model is trained for 50 epochs to achieve 78.37% test accuracy. The confusion matrix and classification report are shown Fig. 14 and Table 7. The summary of model performances on audio cues is shown in Table 8.

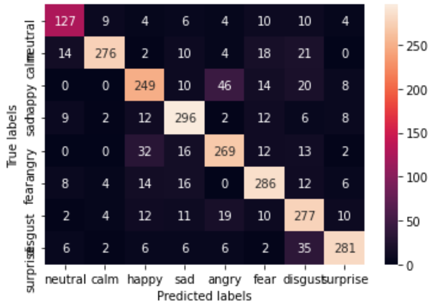


Fig. 13 LSTM model confusion matrix

Table 6: LSTM Model Classification Report

	precision	recall	f1-score	support
Neutral	0.77	0.73	0.75	174
Calm	0.93	0.80	0.86	345
Happy	0.75	0.72	0.73	347
Sad	0.80	0.85	0.82	347
Angry	0.77	0.78	0.78	344
Fear	0.79	0.83	0.81	346
Disgust	0.70	0.80	0.75	345
Surprise	0.80	0.82	0.85	344
Accuracy			0.8	2592
Macro Avg	0.80	0.79	0.79	2592
Weighted Avg	0.80	0.80	0.80	2592

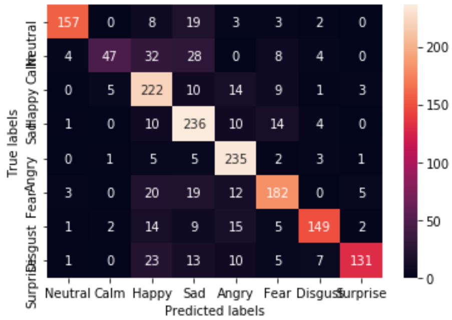


Fig. 14 CNN model for TESS confusion matrix

Table 7: CNN Model Classification Report

	precision	recall	f1-score	support
Neutral	0.94	0.84	0.87	192
Calm	0.85	0.38	0.53	123
Happy	0.66	0.84	0.74	264
Sad	0.70	0.86	0.77	275
Angry	0.79	0.93	0.85	252

Fear	0.80	0.76	0.78	241
Disgust	0.88	0.76	0.81	197
Surprise	0.92	0.69	0.79	190
Accuracy			0.78	1734
Macro Avg	0.82	0.75	0.77	1734
Weighted Avg	0.80	0.78	0.78	1734

Table 8: Audio Cue Model Performances

Method	Accuracy (%)
LSTM	79.51
CNN	78.37

4-4- Experiments on Text Cues

For text emotion recognition we have used a BERT transformer with the datasets mentioned in the above section. BERT uses an attention mechanism to learn the contextual relationship between words in sentences. The network is trained with a learning rate of 0.00002 and batch size of 6 for 2 epochs. We used the Ktrain library to implement a BERT transformer. For textual cue emotion recognition, we achieved 82.26% test accuracy. The confusion matrix and classification report are displayed in Fig. 14 and Table 9.

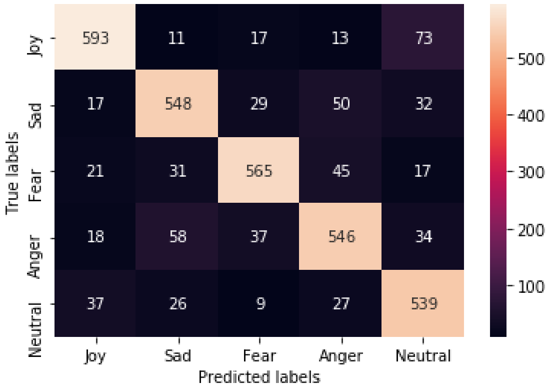


Fig. 14 Textual cue confusion matrix

Table 9: Textual cue Classification Report

	precision	recall	f1-score	support
Joy	0.86	0.84	0.85	707
Sad	0.81	0.81	0.81	676
Fear	0.86	0.83	0.85	679
Anger	0.80	0.79	0.79	693
Neutral	0.78	0.84	0.81	638
Accuracy			0.82	3393
Macro Avg	0.82	0.82	0.82	3393
Weighted Avg	0.82	0.82	0.82	3393

The proposed framework provides average accuracy of 79.48% for emotional distress detection from personalized affective cues when tested on e'NTERFACE dataset. The accuracy metrics of all the modalities are displayed in Table 10.

Table 10: Audio Cue Model Performances

Method	Accuracy (%)
Video cues	77.24
Audio cues	78.94
Text cues	82.36
Hierarchical Fusion	79.48

5- Conclusions

In this paper, we propose a hierarchical weighted framework for emotional distress detection using personalized affective cues i.e., facial expressions, audio signals and textual data. For the face affective state detection, we have utilized a multiple model approach with different datasets. We have also taken the advantage of transfer learning with ResNet50 to extract high-level features from video frames. Facial affective state recognition is challenging, considering variability in dataset and emotion demonstration by an individual. The second cue is an audio cue. We extracted low-level acoustic features of audio and stored them in a matrix to process it using two different approaches. We employed both CNN and LSTM capabilities separately for audio cues, which proved to be better than the combined CNN+LSTM approach. Text cue is last but not the least. For textual cues, we take advantage of the transferred learning using the BERT transformer to extract detailed word embeddings. All the cues are fused to generate a final affective state of the person. Finally, emotional distress is detected by analyzing the observed emotions. Experiments on multiple challenging datasets validate that our method is efficient and viable. For future work, the proposed algorithms can be deployed on smartphones for the self-assessment of emotional distress before consulting the clinical practitioners. The proposed algorithm can be combined with a clinical distress assessment questionnaire for effective results. Combining the results of our proposed architecture with clinical evidence will help in diagnosing mental disorders in the early stage.

Acknowledgements

I would like to acknowledge my supervisor Dr. Rekha Sugandhi, MIT ADT University, Pune, India for her

valuable support, insights and significant contribution in this research work.

References

[1] Gu Simeng, Wang Fushun, Patel Nitesh P., Bourgeois James A., Huang Jason H, “A Model for Basic Emotions Using Observations of Behavior in Drosophila,” Frontiers in Psychology, vol. 10, 2019, pp.781.

[2] Rana R, Latif S, Gururajan R, Gray A, Mackenzie G, Humphris G, Dunn J, “Automated screening for distress: A perspective for the future,” The European Journal of Cancer Care, vol. 28(4), 2019, pp. 1-13.

[3] Riba, M. B. et al., “Distress Management Version 3.2019 ,” NCCN Clinical Practice Guidelines in Oncology, Journal of the National Comprehensive Cancer Network, vol.17(10), 2019, pp.1229–1249.

[4] A. Mehrabian and S.R. Ferris, “Inference of attitudes from nonverbal communication in two channels,” International Journal of consulting psychology, vol. 31(3), 1967, pp. 248–252.

[5] T. Thomas, M. Domínguez and R. Ptucha, "Deep independent audio-visual affect analysis," in 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2017, pp. 1417-1421.

[6] J. A. Miranda, M. F. Canabal, J. M. Lanza-Gutiérrez, M. P. García and C. López-Ongil, "Toward Fear Detection using Affect Recognition," in 2019 XXXIV Conference on Design of Circuits and Integrated Systems (DCIS), November 2019, pp. 1-4.

[7] Lang He, Dongmei Jiang, and Hichem Sahli. “Multimodal depression recognition with dynamic visual and audio cues,” Proc. 2015 International Conference on Affective Computing and Intelligent Interaction, IEEE Computer Society, USA, 2015, pp. 260–266.

[8] Guangxia Xu, Weifeng Li, Jun Liu, “A social emotion classification approach using multi-model fusion,” Future Generation Computer Systems, vol. 102, 2020, pp. 347-356.

[9] Zeinab Farhoudi, Saeed Setayeshi, “Fusion of deep learning features with mixture of brain emotional learning for audio-visual emotion recognition,” Speech Communication,” vol. 127, 2021, pp. 92-103.

[10] Do LN., Yang HJ., Nguyen, HD. et al. “Deep neural network-based fusion model for emotion recognition using visual data,” Journal of Supercomputing, vol.77, 2021, pp. 1-18.

[11] Neha Jain, Shishir Kumar, Amit Kumar, Pourya Shamsolmoali, Masoumeh Zareapoor, “Hybrid deep neural networks for face emotion recognition,” Pattern Recognition Letters, vol. 115, 2018, pp. 101-106.

[12] Heysem Kaya, Furkan Grpnar, and Albert Ali Salah, “Video-based emotion recognition in the wild using deep transfer learning and score fusion,” Image Vision Computing, vol. 65, 2017, pp. 66–75.

[13] Shiqing Zhang, Xin Tao, Yuelong Chuang, Xiaoming Zhao, “Learning deep multimodal affective features for spontaneous speech emotion recognition, Speech Communication,” vol. 127, 2021, pp. 73-81.

[14] Jingwei Yan, Wenming Zheng, Zhen Cui, Chuangao Tang, Tong Zhang, Yuan Zong, “Multi-cue fusion for emotion recognition in the wild,” Neurocomputing, vol. 309, 2018, pp. 27-35.

[15] Ilyes Bendjoudi, Frederic Vanderhaegen, Denis Hamad, Fadi Dornaika, “Multi-label, multi-task CNN approach for context-based emotion recognition,” Information Fusion, November 2020, in press.

[16] Man Hao, Wei-Hua Cao, Zhen-Tao Liu, Min Wu, Peng Xiao, “Visual-audio emotion recognition based on multi-task and ensemble learning with multiple features,” Neurocomputing, vol. 391, 2020, pp. 42-51.

[17] Tzirakis, P., Trigeorgis, G., Nicolaou, M.A., Schuller, B., Zafeiriou, S., “End-to-End Multimodal Emotion Recognition Using Deep Neural Networks,” IEEE Journal of Selected Topics in Signal Processing, vol. 11, 2017, pp. 1301-1309.

[18] N. Majumder, D. Hazarika, A. Gelbukh, E. Cambria, S. Poria, “Multimodal sentiment analysis using hierarchical fusion with context modeling, Knowledge-Based Systems,” vol. 161, 2018, pp. 124-133.

[19] Soujanya Poria, Erik Cambria, Amir Hussain, Guang-Bin Huang, “Towards an intelligent framework for multimodal affective data analysis,” Neural Networks, vol. 63, 2015, pp. 104-116.

[20] Yaxiong Ma, Yixue Hao, Min Chen, Jincai Chen, Ping Lu, Andrej Košir, “Audio-visual emotion fusion (AVEF): A deep efficient weighted approach,” Information Fusion, vol. 46, 2019, pp. 184-192.

[21] Jie Guo, Bin Song, Peng Zhang, Mengdi Ma, Wenwen Luo, Junmei lv, “Affective video content analysis based on multimodal data fusion in heterogeneous networks,” Information Fusion, vol. 51, 2019, pp. 224-232.

[22] Soujanya Poria, Erik Cambria, Newton Howard, Guang-Bin Huang, Amir Hussain, “Fusing audio, visual and textual clues for sentiment analysis from multimodal content,” Neurocomputing, vol. 174, Part A, 2016, pp. 50-59.

[23] F. Noroozi, M. Marjanovic, A. Njegus, S. Escalera and G. Anbarjafari, "Audio-Visual Emotion Recognition in Video Clips," IEEE Transactions on Affective Computing, vol. 10, 2019, pp. 60-75.

[24] S. Zhang, S. Zhang, T. Huang, W. Gao and Q. Tian, "Learning Affective Features with a Hybrid Deep Model for Audio–Visual Emotion Recognition," IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, 2018, pp. 3030-3043.

[25] Li, R., Liu, Z., “Stress detection using deep neural networks.,” BMC Medical Informatics and Decision Making vol. 20, 2020, pp. 285.

[26] P. Bobade and M. Vani, "Stress Detection with Machine Learning and Deep Learning using Multimodal Physiological Data," in 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), 2020, pp. 51-57.

[27] Zhang, H., Feng, L., Li, N., Jin, Z., & Cao, L., “Video-Based Stress Detection through Deep Learning.,” Sensors, vol. 20(19), 2020, pp. 5552.

[28] I. J. Goodfellow et al., “Challenges in representation learning: A report on three machine learning contests,” Neural Networks, Special Issue on Deep Learning of Representations, vol. 64, 2015, pp. 59-63.

[29] Lyons, Michael, Kamachi, Miyuki, & Gyoba, Jiro, “The Japanese Female Facial Expression (JAFFE) Dataset,” Zenodo. 1998.

[30] Lucey et al., "The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified

expression," in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, 2010, pp. 94-101.

[31] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, 2016.

[32] Mehmet Berkehan Akçay, Kaya Oğuz, “Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers, Speech Communication,” vol. 116, 2020, pp. 56-76.

[33] Zheng, F., Zhang, G. & Song, Z., “Comparison of different implementations of MFCC,” Journal of Computer Science & Technology, vol.16, 2001, pp. 582–589.

[34] Livingstone SR, Russo FA, “The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English,” PLoS ONE vol 13(5), 2018.

[35] Pichora-Fuller, M. Kathleen; Dupuis, Kate, "Toronto emotional speech set (TESS)", Scholars Portal Dataverse, V1, 2020.

[36] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova; “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, June 2019.

[37] I. Mureşan, A. Stan, M. Giurgiu and R. Potolea, "Evaluation of sentiment polarity prediction using a dimensional and a categorical approach," in 7th Conference on Speech Technology and Human - Computer Dialogue (SpeD), 2013, pp. 1-6.

[38] Diman Ghazi, Diana Inkpen & Stan Szpakowicz, “Detecting Emotion Stimuli in Emotion-Bearing Sentences”. in 16th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2015), Cairo, Egypt.

[39] Li Yanran, Su Hui, Shen Xiaoyu, Li Wenjie, Cao Ziqiang, Niu Shuzi; “DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset,” in Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers),2017,nov,Asian Federation of Natural Language Processing,Taipei, Taiwan,Pages pp. 986-995.

[40] M. Wöllmer, M. Kaiser, F. Eyben, B. Schuller, G. Rigoll, “LSTM-Modeling of continuous emotions in an audiovisual affect recognition framework,” Image and Vision Computing, vol. 31, 2013, pp. 153-163.

[41] S. Chen and Q. Jin. “Multi-Modal Dimensional Emotion Recognition Using Recurrent Neural Networks”, Proc. 5th International Workshop on Audio/Visual Emotion Challenge. AVEC ’15. Brisbane, Australia: Association for Computing Machinery, 2015, pp. 49-56.

Journal of Information Systems and Telecommunication
Vol.10, No.2, April-June 2022, 102-110

http://jist.acecr.org
ISSN 2322-1437 / ESSN:2345-2773

Smart Pre-Seeding Decision Support System for Agriculture

Ahmed Wasif Reza[1*], Kazi Saymatul Jannat[1], MD. Shariful Islam [1], Surajit Das Barman[1]

[1].Department of Computer Science and Engineering, East West University, Dhaka-1212, Bangladesh

Received: 06 Feb 2021 / Revised: 18 Oct 2021/ Accepted: 09 Dec 2021
DOI:

Abstract

In recent years, the Internet of Things (IoT) brings a new dimension for establishing a precision network connectivity of sensors, especially in the agriculture and farming industry, medical, economic, and several sectors of modern society. Agriculture is an important area for the sustainability of mankind engulfing manufacturing, security, and resource management. Due to the exponential diminishing of the resources, innovative techniques to support the subsistence of agriculture and farming. IoT aims to extend the use of internet technology to a large number of distributed and connected devices by representing standard and interoperable communication protocols. This paper brings up a solution by IoT, presents the design and implementation of a smart pre-seeding decision support system for agricultural modernization. This project is accomplished by understanding the real-time circumstances in the agriculture field using wireless technology that highlighted the features including pH and temperature sensors, hardware, mobile application, system's frontend, and backend analysis, and stores the extracted information in the cloud using IoT. The system is made up of frontend data acquisition, data transmission, data processing, and reception, and is also experimentally validated to find out all possible crops that can be cultivated in a specific land with the required amount of fertilizers as well as the overall crops distribution lists.

Keywords: Agriculture; Internet of Things; Pre-seeding; Decision Support System.

1- Introduction

To fulfill the demand of the modern world, the development and modernization of agriculture are very much on the line with the development of society. In recent times, the concept of the Internet of Things (IoT) in agriculture has become an inevitable trend of agricultural information due to the advancement of internet technology and cloud computing. Technologies like identification and tracking, sensors, actuators, networks, enhanced communication protocols, and distributed intelligence for smart objects are just the most relevant for any serious contribution to the advancement of the IoT [1-4]. Generally, IoT is regarded as a network that interfaces everything with the internet by radio frequency identification (RFID), wired and wireless sensors, GPS, and other information sensing devices.

More than 75 billion devices will be estimated to be connected to the internet by 2025 which may lead to a huge economic impact on the global agriculture and farming markets. In these sectors, there are several issues such as the high capital cost, limitation of farming lands, farmer's poor knowledge on better farming methods, imbalanced utilization of fertilizers, shortage of quality seeds, poor production, dividing the country into zone based on different crops, etc. The rapid growth of IoT-based development technology may lead to inventive and precision agriculture and farming process and can be implemented for managing resources better, monitoring farming fields and crops, economic efficiency, etc. The sensors involved in IoT could be used to track and keep a record of the temperature, humidity, moisture, and pH level in the soil for quantitative productivity with good quality.

IoT-based precision agriculture and farming is convenient for continuous observation and controlling the agriculture industry, especially in remote locations efficiently. Such a smart sensing environment creates a connected network of devices that can share data within them and also be able to take decisions on behalf of a user, and act on the environment to improve its condition. For cultivation, soil testing and their result play an important role in crop selection. Traditional soil testing in the laboratory is time-consuming and inefficient. An

✉ **Ahmed Wasif Reza**
wasif@ewubd.edu

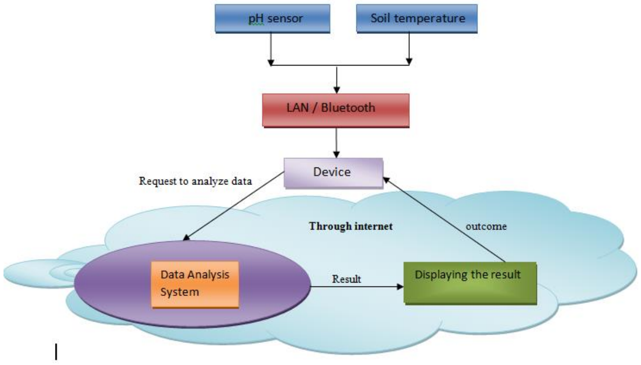
automated technique of measuring soil parameters may overcome this problem. IoT-based pre-seeding decision support using a smart sensing system, also known as a part of “Precision Agriculture”, is an automated technique to identify the possible grains suitable for cultivating in any specific soil. The motivation of this work is to raise awareness among farmers on smart agriculture processes such as how to select an adequate crop list suitable for cultivation on featured land and the number of fertilizers needed for a particular crop that leads to effective productivity and irrigation management.

This paper introduces a cost-effective agriculture pre-seeding decision support system based on the implementation of smart wireless technology featuring pH and temperature sensors, mobile application, system’s frontend, and backend analysis, data acquisition and processing, and information storage in a cloud environment using IoT. The experimental prototype of the proposed system is designed by considering the real-time circumstances of the featured agriculture field. This work aims to demonstrate the accuracy and potential intelligence of the designed system which helps farmers to take decision-making on crop distribution lists suitable for the various cultivating land using continuous monitoring and control of the pH level and soil temperature.

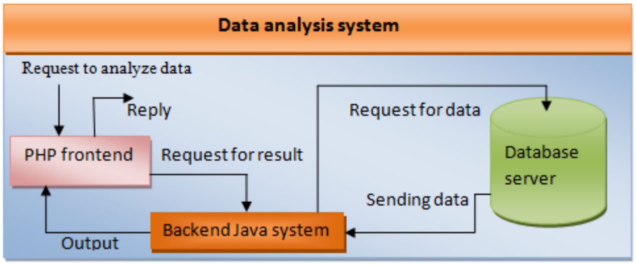
2- Literature Review

Different researchers gave their points of view on the IoT-based smart system for practical and accurate agriculture process applications. Authors in [5] developed a smart remote-controlled robot via ZigBee modules, camera, and actuators with Raspberry Pi for smart irrigation having intelligent decision-making capability utilizing the instantaneous field data. A scalable and feasible reference architecture for water management based on integrating IoT was proposed in [6]. Liang Zhao et al. demonstrated a wireless sensor network middleware for smart agriculture, demand management, crop growth augmentation, optimization, and controlling of agriculture process [7]. An automated intelligent wireless irrigation system was designed to provide an instantaneous feedback control system using LITE mote which effectively observes and controls all the activities of the irrigation system [8]. To enhance efficiency and lower the economic expenditure in the agriculture and farming industry, several smart decision-making and supporting systems based on cloud computing technology were developed [9-17]. Research by Wang and Liu in [16] was mainly focused on the implementation of IoT and cloud computing in agriculture and forestry for data sharing and remote data storage, interactions with farmers, expert consultation and discussion, and household management.

The literature survey mostly focuses on the different quality of service parameters and few of them address the mechanism to achieve an efficient and smart decision-making system for agriculture which may lead to customer dissatisfaction. In Bangladesh, farmers are still depending on the manual recommendation of crops and fertilizers by following a book which is referred from Soil Resource Development Institution (SRDI). This existing method has some limitations such as it recommends only those crops which are generally cultivating in that region and provides the amount of fertilizer that is taken from a previously tested solution. Therefore, it becomes very difficult for a farmer to cultivate crops like rice, wheat, jute, etc. in a small-sized land because all these crops are profitable only for large-sized land. The proposed smart system collects the specific soil parameters by the respective sensors and stores them in the cloud environment for further analysis which assists farmers to decide the suitable crops to be cultivated in a specific land with required fertilizers.



(a)



(b)

Fig. 1 (a) Schematic of system structure, (b) Data analysis system.

3- System Overviews

The pre-seeding monitoring system is designed to precisely select the list of suitable crops for cultivation through the remote monitoring and control of soil conditions. It is a typical IoT system that is built through Arduino Uno R3 and the coordination is adopted via

Arduino Bluetooth Module (JY-MCU). The system is designed for collecting soil information through frontend data acquisition, data transmission, data processing, and reception to guide farmers to find out all possible grains for any specific soil. The structure of the system is shown in Fig. 1.

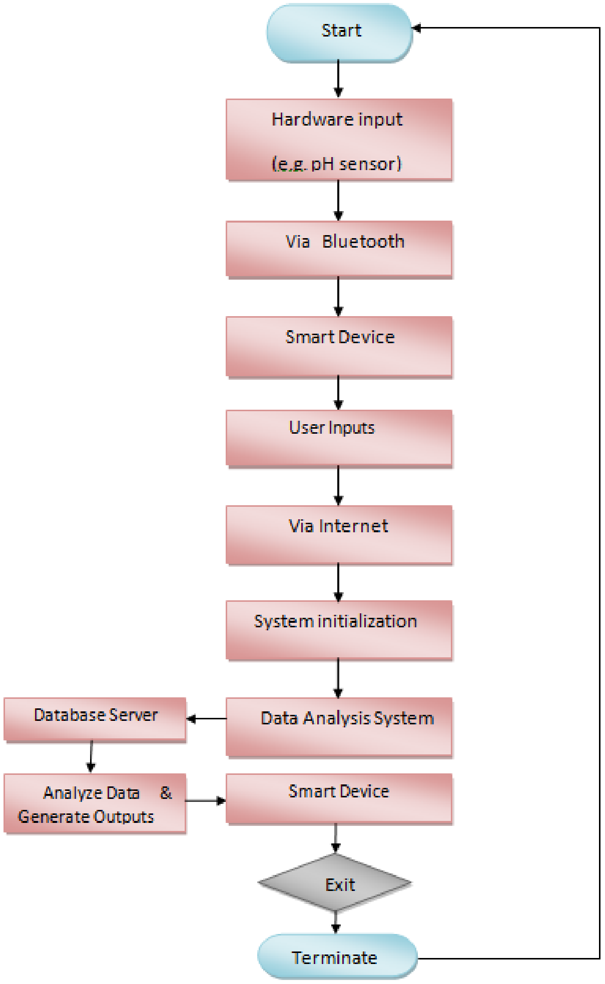


Fig. 2 Design flow of the IoT-based pre-seeding decision support system.

Fig. 2 describes the design flow of the system. In this system, the hardware section extracts inputs from the pH and temperature sensors and transmits them to a smart device (mobile, laptop, etc.) via Bluetooth connection. After receiving inputs, the user gives other necessary inputs manually through a smart device using the internet. The system will initialize all the parameters and transfer them to the data analysis system which is connected with a database server. The system will start processing after receiving all inputs and related data from the database server. During the process, the system will generate all possible desired outputs based on conditions, calculations, and comparison studies. Later, all the processed outputs will be sent back to the user's device using the internet.

4- Methodology and Experimental Setup

The initial design step of the experimental prototype of the pre-seeding system includes the fabrication of the hardware and combines it with the sensors and smart devices for data extraction. In the hardware structure of the system shown in Fig. 3, the Analog pH sensor is connected with Arduino Uno R3 at the Power pin 5V, GND, and Analog input pin A0. Pin RXD/TXD of the Bluetooth module (JY-MCU) is connected with the TXD/RXD pin of the Arduino module and vice-versa. After receiving the value from the pH sensor, the Arduino transmits that extracted value to the smart device using the Bluetooth module. The experimental setup and hardware setup are depicted in Fig. 4.

At the beginning of the processing, the system is initialized by the inputs taken from the hardware, sensors, and users (e.g., location information, soil information, and conditions) through a smart device. Later for final processing, all that information is transferred to the section of the data analysis system via the internet. The data analysis system is developed using java and PHP platforms, and connected with the database server. Fig. 5 illustrates the flow of the processing algorithm used in the system. It goes as in the following process:

1) The initial step of processing is to identify the location. After initializing, the system requests to the database server for all possible mapping units that are presented on that Mouza which is known as the lowest single-area revenue collection.
2) The next step is to identify the soil. At the beginning of this process, the system fetches all soils according to selected mapping units. Then the system will filter soils according to land type information given by the user. Later on, the system selects those soils which are fulfilling two conditions such as water removal condition from the surface and soil consistency at the same time. If the system cannot find any related soil with the above conditions, it will look for a single condition to fulfill which is either water removal condition from the surface or soil consistency. If one of these conditions is accepted, the system collects soil types from the database server. And if none of these have been selected then the system continues with the soils which have been selected according to land type.
3) After selecting the soil series, the system gathers all possible crops information according to irrigation type such as crop name, suitable land type for cultivating, and calculate fertilizers according to crop and recommend it. The system is also capable of

showing the crop distribution lists. And at the same time, it measures whether the pH value is suitable for the crop or not, and can provide suggestions based on pH value.

4) Lastly, the system stores all the processed data and displays it in the smart device through the internet as per user requirements which assists the user to select the suitable crop for cultivation on a particular land.

The exact fertilizer nutrient required for a recommended soil and crop can be stated as

$$F_r = \left(\frac{U_f - C_i}{C_S}\right) \times (S_t - L_S) \qquad (1)$$

where,

F_r = Fertilizer nutrient required for given soil test value,

U_f = Upper limit of the recommended fertilizer nutrient for the respective soil test value interpretation (STVI) class,

C_i = Units of class intervals used for fertilizer nutrient recommendation,

C_S = Units of class intervals used for STVI class,

S_t = Soil test value, and

L_S = Lower limit of the soil test value within STVI class

To obtain the adequate result, the user needs to provide the following information as input to the system:

a) The land type is based on what becomes the scene of the land during the rainy season, whether it goes underwater or not, and if it goes underwater then how much it goes.

b) The soil consists of the land measured by pressing two fingers on the soil.

The water removal condition of the land measuring by when it is possible to cultivate.

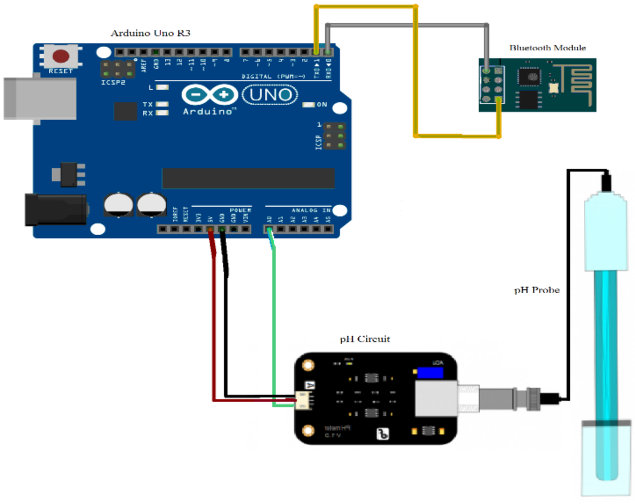
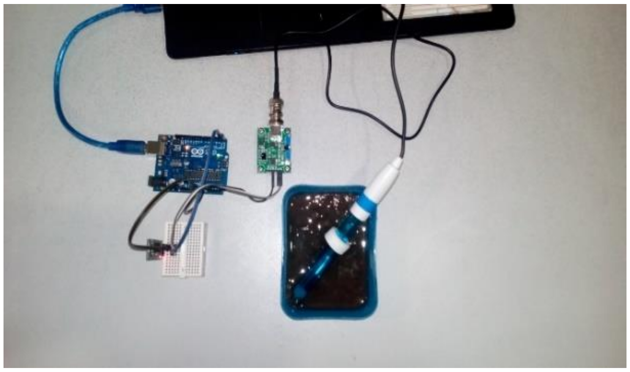
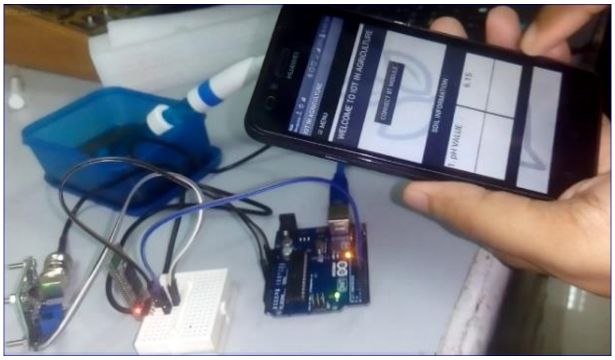


Fig. 3 Circuit diagram of the hardware units.



(a)



(b)

Fig. 4 (a) Hardware setup, (b) Experimental setup of the system.

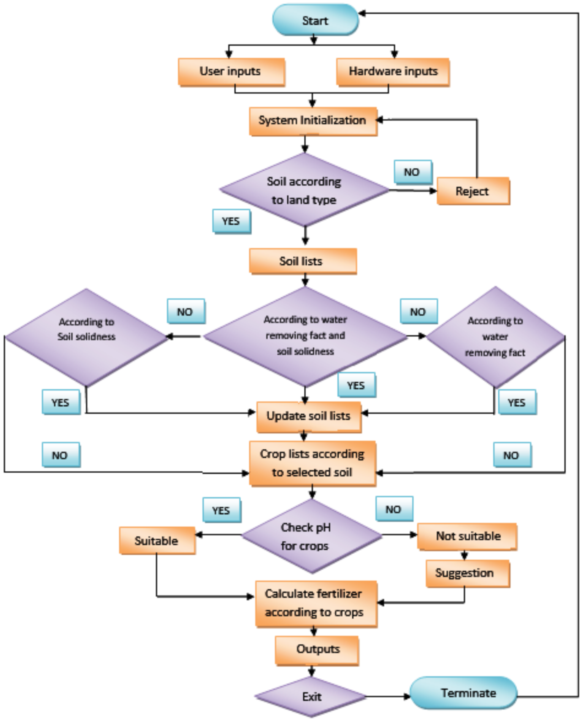
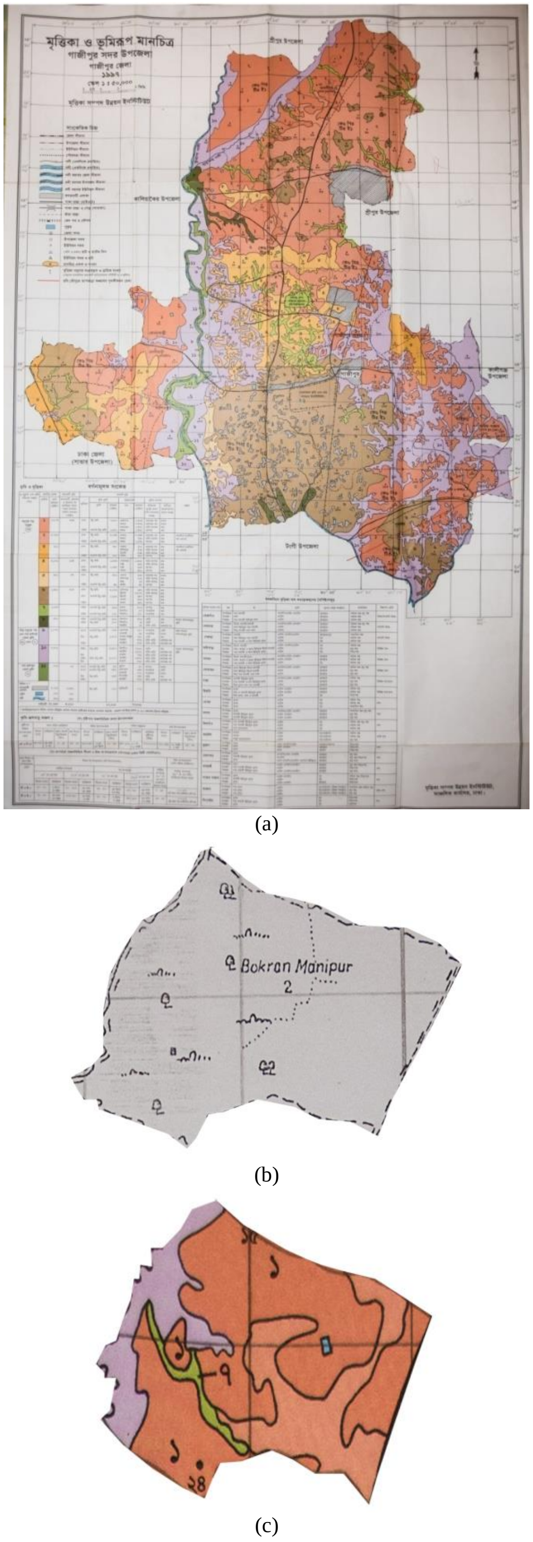

Start
User inputs
Hardware inputs
System Initialization
Soil according to land type
NO
Reject
YES
Soil lists
According to Soil solidness
NO
According to water removing fact and soil solidness
NO
According to water removing fact
YES
YES
YES
Update soil lists
NO
NO
Crop lists according to selected soil
YES
NO
Suitable
Check pH for crops
Not suitable
Suggestion
Calculate fertilizer according to crops
Outputs
Exit
Terminate

Fig. 5 Flow of the processing algorithm of IoT-based pre-seeding system.



(a)

(b)

(c)

Fig. 6 Mapping units of (a) Gazipur Sadar Upazila, (b) Bokran Manipur with Mouza boundary, and (c) Bokran Manipur with mapping units.

5- Results and Discussion

The main objective of the project is to implement the IoT concept in the agriculture field for making the pre-seeding decision process simple. As a Mouza, the Bokran Manipur area under the Mirzapur union of Gazipur Sadar Upazilla in Bangladesh with mapping units: map-1, map-2, map-7, and map-10 were selected, as shown in Fig. 6. The initial value of Nitrogen (N), Phosphorus (P), and Potassium (K) were selected as 0.25, 15, and 0.34, respectively. The pH level of 6.8 to 7.2 is considered neutral and suitable for the land. After updating the values, we got all possible mapping units that are map-1, map-2, map-7, and map-10 according to the union and Mouza, given in Table 1. High land and crumbly soil with immediate water-removing facts were chosen. The designed system can filter the soil series concerning the selected land type listed in Table 2. Table 3 shows the filtered soil based on the "advanced" water removal condition and "crumbly" soil consistency provided by the user. Now, based on the selected soil type (without irrigation), the designed system provides a suggestion about the suitable crop lists (Rabi, Kharif 1, Kharif 2, one-year and Multi-year crops) with recommended fertilizers which are presented in Tables 4 to 8, respectively. Now, based on the selected soil type (without irrigation) the designed system provides the suggestion about the suitable crop lists (Rabi, Kharif 1, Kharif 2, One-year and Multi-year crops) with recommended fertilizers which are presented in Tables 4 to 8, respectively. From the soil pH analyses results, it is observed that Rabi, Kharif 1, Kharif 2 and one-year crops are suitable to grow at a pH value of 7.2 and multi-year crops are suitable to cultivate in the high land of having pH value equal to 6.8. Compared to Rabi and Kharif 1, Kharif 2 class crops require a large amount of Urea, TSP, and KCI per hectare to promote NPK for crop cultivation. Similarly, high quantities of fertilizers are also required for cultivating One-year crops compared to Multi-year crops.

Table 1: All possible soils according to the mapping units

Map units	***Soil series***				
	High land	Medium high land	Medium low land	Low land	Very low land
Map no – 1	Tejgaon, Belab, Noadda	Kolma	–	–	–
Map no – 2	Gerua, Salna	Kolma	–	–	–
Map no – 7	–	Kolma, Khilgaon	Khilgaon	–	–
Map no – 10	–	–	–	Khilgaon, Korail, Kajla	Korail, Kajla

Table 2: Filtered soil with respect to land type

Map units	***Land types***	***Soil series***
Map no – 1	High land	Tejgaon, Belab, Noadda
Map no – 2	High land	Gerua, Salna

Table 3: Filtered soil with water removal condition from the surface and soil consistency

Soil series	***Water removal condition from the surface***	***Drainage system***	***Soil type***	***Soil solidness***
Tejgaon	Immediate	Medium good	Loam	Crumbly
Belab	Immediate	Medium good	Loam	Crumbly
Gerua	Immediate	Medium good	Loam	Crumbly

Table 4: List of Rabi crops based on the selected soil type (without irrigation)

Crop name	***pH status***	***Suitable land***	***Fertilizer (Kg/hectare)***		
			N (Urea)	***P (TSP)***	***K (KCl)***
Mustard	7.2 (Neutral)	High land	104.56	8.867	8.867
Masculine	7.2 (Neutral)	High land	13.406	40.03	2.333
Mung bean	7.2 (Neutral)	High land	16.087	100.067	5.6

Table 5: List of Kharif 1 crops based on the selected soil type (without irrigation)

Crop name	***pH status***	***Suitable land***	***Fertilizer (Kg/hectare)***		
			N (Urea)	***P (TSP)***	***K (KCl)***
Red spinach	7.2 (Neutral)	High land	72.391	60.040	4.667
Chili	7.2 (Neutral)	High land	210.140	15.400	2.391
Kakrol	7.2 (Neutral)	High land	53.623	100.067	4.667
Brinjal	7.2 (Neutral)	High land	174.275	140.093	25.200
Lady’s finger	7.2 (Neutral)	High land	85.797	110.073	9.800
Ground-nut	7.2 (Neutral)	High land	26.812	90.060	7.933
Bona aus Rice	7.2 (Neutral)	High land	40.217	30.033	9.600

Table 6: List of Kharif 2 crops based on the selected soil type (without irrigation)

Crop name	***pH status***	***Suitable land***	***Fertilizer (Kg/hectare)***		
			N (Urea)	***P (TSP)***	***K (KCl)***
Red spinach	7.2 (Neutral)	High land	72.391	60.040	4.667
Masculine	7.2 (Neutral)	High land	13.406	40.027	2.333
Mung bean	7.2 (Neutral)	High land	16.087	100.067	5.6

Table 7: List of One-year crops based on the selected soil type (without irrigation)

Crop name	***pH status***	***Suitable land***	***Fertilizer (Kg/hectare)***		
			N (Urea)	***P (TSP)***	***K (KCl)***
Sugarcane	7.2 (Neutral)	High land	158.188	200.133	25.200
Banana	7.2 (Neutral)	High land	120.652	100.067	20.067
Ginger	7.2 (Neutral)	High land	120.652	170.113	17.733
Pineapple	7.2 (Neutral)	High land	201.187	280.187	35
Turmeric	7.2 (Neutral)	High land	88.478	110.073	12.6

Table 8: List of Multi-year crops based on the selected soil type (without irrigation)

Crop name	***pH status***	***Suitable land***	***Fertilizer (Kg/hectare)***		
			N (Urea)	***P (TSP)***	***K (KCl)***
Lemon	6.8 (Neutral)	High land	69.710	50.033	8.867

Table 9: List of Rabi crops based on the selected soil type (with irrigation)

Crop name	***pH status***	***Suitable land***	***Fertilizer (Kg/hectare)***		
			N (Urea)	***P (TSP)***	***K (KCl)***
Wheat	7.2 (Neutral)	High land	107.246	110.073	12.6
Corn	7.2 (Neutral)	High land	107.246	110.073	13.067
Potato	7.2 (Neutral)	High land	120.652	100.067	18.667
Mustard	7.2 (Neutral)	High land	104.565	100.067	8.867
Sunflower	7.2 (Neutral)	High land	88.478	100.067	8.4
Cabbage	7.2 (Neutral)	High land	174.275	140.093	25.2
Red spinach	7.2 (Neutral)	High land	72.391	60.04	4.667
Chili	7.2 (Neutral)	High land	210.14	15.4	2.391
Lady’s finger	7.2 (Neutral)	High land	85.797	110.073	9.8
Brinjal	7.2 (Neutral)	High land	174.275	140.093	25.2
Cotton	7.2 (Neutral)	High land	99.203	110.073	15.4
Carrot	7.2 (Neutral)	High land	120.652	130.087	14
Cauliflower	7.2 (Neutral)	High land	131.377	170.113	18.667
Groundnut	7.2 (Neutral)	High land	26.812	90.06	7.993

Table 10: List of Kharif 1 crops based on the selected soil type (with irrigation)

Crop name	*pH status*	*Suitable land*	*Fertilizer (Kg/hectare)*		
			N (Urea)	*P (TSP)*	*K (KCl)*
Red spinach	7.2 (Neutral)	High land	26.812	90.06	7.933
Corn	7.2 (Neutral)	High land	107.246	110.073	13.067
Chili	7.2 (Neutral)	High land	210.14	15.4	2.391
Kakrol	7.2 (Neutral)	High land	53.623	100.067	4.667
Brinjal	7.2 (Neutral)	High land	174.275	140.093	25.2
Lady's finger	7.2 (Neutral)	High land	85.797	110.073	9.8

Table 11: List of One-year crops based on the selected soil type (with irrigation)

Crop name	*pH status*	*Suitable land*	*Fertilizer (Kg/hectare)*		
			N (Urea)	*P (TSP)*	*K (KCl)*
Sugarca-ne	7.2 (Neutral)	High land	158.188	200.133	25.2
Banana	7.2 (Neutral)	High land	120.652	100.067	20.067
Pineapp-le	7.2 (Neutral)	High land	201.087	280.187	35

Table 12: List of Multi-year crops based on the selected soil type (with irrigation)

Crop name	*pH status*	*Suitable land*	*Fertilizer (Kg/hectare)*		
			N (Urea)	*P (TSP)*	*K (KCl)*
Lemon	6.8 (Neutral)	High land	69.71	50.033	8.867
Betel	6.8 (Neutral)	High land	53.623	150.1	7

Table 13: Crop distribution lists provided by the IoT based pre-seeding system (without irrigation)

Serial no.	*Distribution of crops*
1	Mustard-kharif vegetables/kharif groundnut
2	Fallow-aus-masculine
3	Sugarcane
4	Banana/Ginger/Turmeric
5	Fallow-T.aus

Table 14: Crop distribution lists provided by the IoT based pre-seeding system (with irrigation)

Serial no.	*Distribution of crops*
1	Sugarcane
2	Rabi vegetables/groundnut Kharif vegetables
3	Rabi crops/bona aus
4	Banana/Pineapple
5	Betel/Lemon

Table 15: Comparison of detected key soil parameters for Wheat, Rice, and Sugarcane

Crops	*Soil parameters*	*Designed systems*		
		Spandana et al. [16]	*Badhe et al. [17]*	*Proposed model*
Wheat	*Land type*	Red	-	High land
	pH value	4	5.5-6.5	7.2
	Moisture	50	21-24	-
	Urea (N)	-	-	107.246 Kg/hectare
	TSP (P)	-	-	110.073 Kg/hectare
	KCI (K)	-	-	12. 6 Kg/hectare
Rice	*Land type*	Red	-	High land
	pH value	7	5.5-7	7.2
	Moisture	60	21-37	-
	Urea (N)	-	-	40.217 Kg/hectare
	TSP (P)	-	-	30.033 Kg/hectare
	KCI (K)	-	-	9. 6 Kg/hectare
Sugar-cane	*Land type*	-	-	High land
	pH value	-	5-8.5	7.2
	Moisture	-	28-32	-
	Urea (N)	-	-	158.188 Kg/hectare
	TSP (P)	-	-	200.133 Kg/hectare
	KCI (K)	-	-	25.2 Kg/hectare

The system also provides a suitable list of crops according to the irrigation technique listed in Tables 9 to 12. Similar to the crop lists under the selected soil type (without irrigation), the optimum suitable pH value is between 6.8 – 7.2 for the crops according to irrigation technique. In this case, the selected land type requires a large number of fertilizers per hectare to cultivate Rabi and One-year crops, respectively.

As observed, the proposed pre-seeding system is capable of providing suggestions regards the possible crops to cultivate (with or without irrigation) along with recommended fertilizers as per the selected land type and suitable season. The system can also be capable of providing the crop distribution lists in a sequence for cultivating based on irrigation techniques, which are shown in Table 13 and Table 14, respectively. In contrast with previous research in [16] and [17] on crop recommendation using IoT, the proposed model is capable of effectively recommending land type, specific pH label, and the exact amount of fertilizers required for the three most important crops: Wheat, Rice, and Sugarcane in Bangladesh which is highlighted in Table 15. Compared to those works, the extracted soil pH status using the designed prototype is neutral and all three crops are suitable to cultivate at this value. On requirement, the user also gets to know the number of optimum fertilizers required to boost NPK in the soil which is not analyzed in previous studies.

6- Conclusions

As agriculture is now one of the most important sectors for the growth of the economy in developing countries, technological advances must be utilized to stay aware of the increasing demand of the human populace. In this paper, an IoT-based smart pre-seeding decision support system is developed which realizes intelligent recommendations of suitable crops for the specific farming area. The hardware resources in the agriculture information network are integrated with the IoT sensors and smart devices. To validate the model through experiment, high land and crumbly soil of Bokran Manipur area under the Mirzapur union of Gazipur Sadar Upazilla in Bangladesh were chosen. The experimental test and all observations show its effectiveness in suggesting farmable crops based on the irrigation technique with recommended fertilizers and pH status along with the crop distribution list. It is found that a suitable pH range of 6.8-7.2 is required to grow Rabi, Kharif 1, Kharif 2, One-year, and Multi-year crops depending on the soil irrigation method. The selected soil types based on both without irrigation and with irrigation require a large quantity of Urea, TSP, and KCI fertilizer to grow Kharif 2 and Rabi crops, respectively via elevating the NPK value in the land.
Although this study has made significant progress in some areas, there are still a few shortcomings. The designed prototype requires continuous internet connectivity which is the biggest challenge to implementing the smart agriculture farming concept, especially in rural areas of Bangladesh. Also, farmers in rural areas are less interested to acquire technological knowledge for smart farming and prefer traditional farming. Finally, the designed pre-seeding decision-making system would help farmers to make intelligent decisions on planting, fertilizing, and harvesting crops. In the future, the system could be enhanced to cover large acres of agricultural lands, analyze their moisture level, and be integrated with the website and mobile applications to ensure direct information exchange between farmers and the system. Also, the system can be integrated to check the soil quality and the growth rate of crops.

Acknowledgments

The authors would like to thank Sher-e-Bangla Agricultural University, which is the oldest agricultural institution in Bangladesh and South Asia for providing expert advice and support in data collection, result validation, and conducting this research successfully.

References

[1] K. Raviteja, and M. Supriya, “IoT-Based Agriculture Monitoring System”, Data Engineering and Communication Technology, Vol. 1079, 2020, pp. 473-483.

[2] S. Ratnaparkhi, S. Khan, C. Arya, S. Khapre, P. Singh, M. Diwakar, and A. Shankar, “Smart agriculture sensors in IOT: A review”, in Materials Today: Proceedings, 2020.

[3] P. M. Jyothi, and D. Nandan, “Utilization of the Internet of Things in Agriculture: Possibilities and Challenges”, Soft Computing: Theories and Applications, 2020, pp. 837-848.

[4] M. A. Ferrag, L. Shu, X. Yang, A. Derhab, and L. Maglaras, “Security and Privacy for Green IoT-Based Agriculture: Review, Blockchain Solutions, and Challenges”, IEEE Access, Vol. 8, 2020, pp. 32031-32053.

[5] N. Gondchawar, and R. S. Kawitkar “IoT based Smart Agriculture”, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 5, No. 6, 2016, pp. 838-842.

[6] T. Robles, R. Alcarria, D. Martin, M. Navarro, R. Calero, S. Iglesias, and M. Lopez, "An IoT based reference architecture for smart water management processes", Journal of Wireless Mobile Networks, Ubiquitous Computing and Dependable Applications, Vol. 6, No. 1, 2015, pp. 4-23.

[7] L. Zhao, L. He, X. Jin, and W. Yu, “Design of Wireless Sensor Network Middleware for Agricultural Applications”, in International Conference on Computer and Computing Technologies in Agriculture, 2012, pp. 270-279.

[8] K. R. Anusha, and K. R. Shobha, "Design and Implementation of Wireless Sensor Network for Precision Agriculture", International Journal Scientific Engineering and Applied Science, Vol. 1, 2015, pp. 521-527.

[9] S. A. Jaishetty, and R. Patil, "IoT Sensor Network Based Approach for Agricultural Field Monitoring and Control", International Journal of Research in Engineering and Technology, Vol. 5, No. 06, 2016, pp. 2319-1163.

[10] P. Sanjeevi, and P. Viswanathan, “NUTS scheduling approach for cloud data centers to optimize energy consumption”, Computing, Vol. 99, No. 12, 2017, pp. 1179-1205.

[11] P. Sanjeevi, and P. Viswanathan, “Smart homes IoT techniques for dynamic provision of cloud benefactors”, International Journal of Critical Computer-Based Systems, Vol. 7, No. 3, 2017, pp. 209-224.

[12] G. Kesavan, P. Sanjeevi, and P. Viswanathan, “A 24 hours IoT framework for monitoring and managing home automation”, in International Conference on Inventive Computation Technologies (ICICT), 2016, Vol. 1, pp. 367-371.

[13] M. A. Supreetha, M. R. Mundada, and J. N. Pooja, “Design of a smart water-saving irrigation system for agriculture based on a wireless sensor network for better crop yield”, in International Conference on Communications and Cyber Physical Engineering, 2018, pp. 93-104.

[14] M. A. Mattar, T. K. Z. El-Abedin, A. A. Alazba, and H. M. Al-Ghobari HM, “Soil water status and growth of tomato with partial root-zone drying and deficit drip irrigation techniques”, Irrigation Science, Vol. 38, No. 2, 2020, pp. 163-176.

[15] A. Kapoor, S. I. Bhat, S. Shidnal, and A. Mehra, "Implementation of IoT (Internet of Things) and Image Processing in Smart Agriculture", in International

Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), 2016, pp. 21-26.

[16] K. Spandana, and S. Pabboju, S. "Applications of IoT for Soil Quality," in International Conference on Intelligent Computing and Communication Technologies, 2019, pp. 277-286.

[17] A. Badhe, S. Kharadkar, R. Ware, P. Kamble, and S. Chavan, "IOT based smart agriculture and soil nutrient detection system", International Journal on Future Revolution in Computer Science & Communication Engineering, Vol. 4, No. 4, 2018, pp. 774-777.

[18] H. Z. Wang, G. W. Lin, J. Q. Wang, W. L. Gao, Y. F. Chen, and Q. L. Duan, "Management of big data in the internet of things in agriculture based on cloud computing", applied mechanics and materials, Vol. 548, 2014, pp. 1438-1444.

Optimized kernel Nonparametric Weighted Feature Extraction for Hyperspectral Image Classification

Mohammad Hasheminejad[1*]

[1].Department of Electrical Engineering, University of Jiroft, Kerman, Iran

Received: 13 May 2021/ Revised: 04 Oct 2021/ Accepted: 21 Dec 2021
DOI:

Abstract

Hyperspectral image (HSI) classification is an essential means of the analysis of remotely sensed images. Remote sensing of natural resources, astronomy, medicine, agriculture, food health, and many other applications are examples of possible applications of this technique. Since hyperspectral images contain redundant measurements, it is crucial to identify a subset of efficient features for modeling the classes. Kernel-based methods are widely used in this field. In this paper, we introduce a new kernel-based method that defines Hyperplane more optimally than previous methods. The presence of noise data in many kernel-based HSI classification methods causes changes in boundary samples and, as a result, incorrect class hyperplane training. We propose the optimized kernel non-parametric weighted feature extraction for hyperspectral image classification. KNWFE is a kernel-based feature extraction method, which has promising results in classifying remotely-sensed image data. However, it does not take the closeness or distance of the data to the target classes. Solving the problem, we propose optimized KNWFE, which results in better classification performance. Our extensive experiments show that the proposed method improves the accuracy of HSI classification and is superior to the state-of-the-art HIS classifiers.

Keywords: Feature Extraction; Image Classification; Optimized KNWFE; Hyperspectral; Kernel.

1- Introduction

Hyperspectral image (HSI) classification is widely used in many fields such as agriculture, mineralogy [1], environmental monitoring, and material analysis [2]. An HSI image contains spatial-spectral information, which is the visible and near-infrared, and short-wavelength infrared spectrum, for different locations in an image plane. This image plane is usually obtained by airborne and spaceborne spectrometers [3]. These images have many spectral bands and complex spatial structures containing lots of information. These images typically cover a wide spectral range of frequencies. As a result, each pixel vector is a highly-detailed spectral representative of each captured land cover material. Therefore, since the types of materials on the ground are better identified using HSI images, they can be used in many applications performed via surface analysis. The analysis of HSI involves classification. The goal of classification is to assign a unique class label to each pixel vector.

As an example of HSI classification methods, SVM can be cited [4]. SVM searches an optimal hyperplane to separate the data in a multi-dimensional feature space. Other widely used spectral classification methods include k-nearest-neighbors, maximum likelihood, logistic regression, neural networks [5]. To avoid the computational burden and increase the classification accuracy, it is recommended to use dimensionality reduction techniques [6]. In the past several years, many feature extraction and classification methods have been presented for hyperspectral data [1], [7]. An example of supervised dimensionality reduction is linear discriminant analysis[8]. Besides, non-parametric weighted feature extraction (NWFE) [9], local joint subspace (LJS) detection [4], independent component analysis [10], principal component analysis [11], superpixelwise PCA [12], and semi-supervised discriminant analysis (SDA) [12] are dimensionality reduction methods which are considered by the community. However, due to the unbalance between the limited number of training samples and the high dimensionality of data, HSI classification is still a highly challenging task [13].

In hyperspectral image classification, each pixel is labeled with one of the classes based on its features. SVM is known as a powerful method in HSI classification [14]. Another classifier that is widely used is multinomial

✉ **Corresponding Author**
mhn@ujiroft.ac.ir

logistic regression [15]. This classifier uses the logistic function to provide the posterior probability. In [15], an ensemble multinomial logistic regression-based method is used for HSI classification. An anomalous component extraction framework for the detection of hyperspectral anomalies based on Independent Component Analysis (ICA) and orthogonal subspace imaging (OSP) is proposed in [16]. Kernel-based SVM approaches can offer satisfying performance in HSI classification. Mountrakis et al. showed that using a nonlinear kernel with a local k-nearest neighbor adaptation improves the performance of localized types of SVM approaches [17]. A regularization method is proposed in [18] to address the issue of kernel predetermination. The technique identifies kernel structure through the analysis of unlabeled samples. H. C. Lee et al. proposed an HSI classifier that projects Gabor features of the hyperspectral image into the kernel induced space through composite kernel technique [1]. Representation-based methods such as sparse representation are proven to be promising in pattern recognition. HSI sparse representation classification is based on the assumption that pixels belonging to the same class lie in the same subspace. It is also applied to HSI classification [19], where the representation is performed in a feature space induced by a kernel function. Sparse representation classification is now a popular method in hyperspectral unmixing. Weng et al. used a kernel to map hyperspectral data and library atoms to a suitable space to unmix hyperspectral information [20]. Sparse representation is also used to enhance hyperspectral images [21].

Recently, a variety of deep learning-based algorithms has shown their promising performance in various applications, including HSI classification [22]. Due to the success of deep learning in the field of pattern recognition, it has attracted many researchers in hyperspectral image classification and analysis [23], [24]. In [23], a convolutional neural network (CNN) architecture is proposed for HSI classification. They proposed a 3-D network that uses both spectral and spatial information. To effectively process the border areas in the image, it implemented a border mirroring strategy. The proposed algorithm is implemented on graphical processing units. In [24], a simplified deep neural network is proposed. This network, which is called MugNet, utilizes the relationship between different spectral bands and neighboring pixels. It also generates a convolution kernel using a semi-supervised manner. The application of deep SVM in HSI classification is investigated in [25]. Four kernel functions were used in that study.

However, it is commonly necessary to pre-process that spectral information to use in HSI analysis. This process includes reducing the number of bands using proper techniques. In this case, non-parametric weighted feature extraction (NWFE) has shown promising results in HSI dimension reduction [9]. It is further improved in [26] as KNWFE, taking advantage of the kernel method. In this paper, we try to improve within and between class scattering matrices, correcting data weightings.

The rest of this paper is organized as follows: Section 2 overviews the KNWFE method. In Section 3, we propose our corrections on the KNWFE followed by the performed experiments in Section 4. We conclude in Section 5.

2- Related Work

Most of the time, HSIs are not linearly separable. Therefore kernel methods are used to project the data into a feature space, where the classes are linearly separable. The kernel function is a similarity function that corresponds to an inner product in some expanded feature space. Some popular kernel functions are linear kernel, polynomial kernel and gaussian radial-basis-function (RBF) kernel.

The proposed algorithm is a nonlinear kernel-mode based on the nonparametric weighted feature extraction (NWFE) method [26]. NWFE is a nonparametric method for high-dimensional multi-class pattern recognition problems. This algorithm is based on a non-parametric expression of the scatter matrix. The steps of this algorithm are to first calculate the Euclidean distance between each sample pair and place it in a matrix called the distance matrix. Then the weights matrix is calculated using the distance matrix. The weighted mean matrix is then calculated by putting different weights on every sample. Then, the distance between samples and their weighted means is calculated, as their closeness to the boundary. Finally, nonparametric between-class and within-class scatter matrices are defined, to put large weights on the samples close to the boundary and deemphasize samples far from the boundary. These matrices are defined respectively as [26]:

$$S_b^{NW} = \sum_{i=1}^{L} P_i \sum_{\substack{j=1 \\ j\neq i}}^{L} \sum_{\ell=1}^{N_i} \frac{\lambda_\ell^{(i,j)}}{N_i} \left(x_\ell^{(i)} - M_j\left(x_\ell^{(i)}\right)\right) \times \left(x_\ell^{(i)} - M_j\left(x_\ell^{(i)}\right)\right)^T$$

$$S_w^{NW} = \sum_{i=1}^{L} P_i \sum_{\ell=1}^{N_i} \frac{\lambda_\ell^{(i,j)}}{N_i} \left(x_\ell^{(i)} - M_j\left(x_\ell^{(i)}\right)\right) \times \left(x_\ell^{(i)} - M_j\left(x_\ell^{(i)}\right)\right)^T$$

where $\lambda_\ell^{(i,j)}$ is scatter matrix weight and is defined by:

$$\lambda_\ell^{(i,j)} = \frac{dist\left(x_\ell^{(i)}, M_j\left(x_\ell^{(i)}\right)\right)^{-1}}{\sum_{t=1}^{N_i} dist\left(x_t^{(i)}, M_j\left(x_t^{(i)}\right)\right)^{-1}}$$

with $M_j\left(x_\ell^{(i)}\right)=\sum_{k=1}^{N_j}\omega_{\ell k}^{(i,j)}x_k^{(j)}$, that denoted the weighted mean concerning $x_\ell^{(i)}$ in class j, $dist(A,B)$ the distance between A and B, and

$$\omega_{\ell k}^{(i,j)}=\frac{dist\left(x_\ell^{(i)},x_k^{(j)}\right)^{-1}}{\sum_{t=1}^{N_j}dist\left(x_\ell^{(i)},x_t^{(j)}\right)^{-1}}$$

Despite that NWFE has better performance than LDA, it is still linear. The KNWFE method, a kernel-based nonlinear version of the NWFE, is presented to derive the non-Gaussian data feature [26]. In this method, $x_\ell^{(i)}$ in the scatter matrices is replaced by $\varphi\left(x_\ell^{(i)}\right)$, where $\varphi(.)$ is a kernel function.

2-1- Kernel Nonparametric Weighted Feature Extraction

The strategy of kernel-based methods is to map data from the original space to a higher-dimensional Hilbert space, where the data are expected to be more separable in this space. The kernel is an N×N matrix, where N is the total number of samples. In KNWFE, a weight matrix is firstly defined, based on data.

$$\lambda_l^{(i,j)}=\frac{\left[K_{ll}^{(i,i)}+(W^{(i,j)}K^{(j,j)}W^{(i,j)^T})_{ll}-2(K^{(i,j)}W^{(i,j)^T})_{ll}\right]^{-1/2}}{\sum_{t=1}^{N_i}\left[K_{tt}^{(i,i)}+(W^{(i,j)}K^{(j,j)}W^{(i,j)^T})_{tt}-2(K^{(i,j)}W^{(i,j)^T})_{tt}\right]^{-1/2}} \quad (1)$$

Where l represents a datum, $i=1,2,\dots,L$, $j=1,2,\dots,L$, L is the number of classes, $K^{(i,j)}$ is a part of kernel matrix, K, and $W^{(i,j)}$ which is shown in (2). Matrix $\Lambda^{(i,j)}$, is then defined as in (3) which is used in the process [26].

$$W^{(i,j)}=\begin{bmatrix}w_{11}^{(i,j)} & \cdots & w_{1N_j}^{(i,j)}\\ \vdots & \ddots & \vdots\\ w_{N_i1}^{(i,j)} & \cdots & w_{N_iN_j}^{(i,j)}\end{bmatrix} \quad (2)$$

$$\Lambda^{(i,j)}=diag\left\{\frac{\lambda_1^{(i,j)}}{N_i},\dots,\frac{\lambda_{N_i}^{(i,j)}}{N_i}\right\} \quad (3)$$

Where $w_{lk}^{(i,j)}$ is defined in (4), and N and N_i are total number of data, and number of data in class i, respectively.

$$w_{lk}^{(i,j)}=\frac{(K_{ll}^{(i,i)}+K_{kk}^{(j,j)}-2K_{lk}^{(i,j)})^{-1}}{\sum_{t=1}^{N_j}(K_{ll}^{(i,i)}+K_{tt}^{(j,j)}-2K_{lt}^{(i,j)})^{-1}} \quad (4)$$

To obtain a transformation matrix, it is firstly needed to calculate the two matrices W and B (equations (5)-(12)).

$$W=W_1-W_2-W_2^T+W_3 \quad (5)$$

$$W_1=diag\left\{P_1\Lambda^{(1,1)},\dots,P_L\Lambda^{(L,L)}\right\} \quad (6)$$

$$W_2=diag\left\{P_1\Lambda^{(1,1)}W^{(1,1)},\dots,P_L\Lambda^{(L,L)}W^{(L,L)}\right\} \quad (7)$$

$$W_3=diag\{P_1W^{(1,1)^T}\Lambda^{(1,1)}W^{(1,1)},\dots\\ ,P_LW^{(L,L)^T}\Lambda^{(L,L)}W^{(L,L)}\} \quad (8)$$

$$B=B_1-B_2-B_2^T+B_3 \quad (9)$$

$$B_1=diag\left\{P_1\sum_{j=1}^{L}\Lambda^{(1,j)},\dots,P_L\sum_{j=1}^{L}\Lambda^{(L,j)}\right\} \quad (10)$$

$$B_2=\begin{bmatrix}P_1\Lambda^{(1,1)}W^{(1,1)} & \cdots & P_1\Lambda^{(1,L)}W^{(1,L)}\\ \vdots & \ddots & \vdots\\ P_L\Lambda^{(L,1)}W^{(L,1)} & \cdots & P_L\Lambda^{(L,L)}W^{(L,L)}\end{bmatrix} \quad (11)$$

$$B_3=\sum_{i=1}^{L}P_i diag\{\\ W^{(i,1)^T}\Lambda^{(i,1)}W^{(i,1)},\dots,W^{(i,L)^T}\Lambda^{(i,L)}W^{(i,L)}\} \quad (12)$$

In the above equations, P_i is the probability of ith class. The following steps are taken to obtain the transformation matrix (A) [26].

If the transformation matrix is derived according to Fisher's relationship as follows (Equation (13)), then it is necessary to use the decomposition of the eigenvalue and eigenvector to obtain P and U:

$$A=PU \quad (13)$$

Where P is the eigenvector of the kernel matrix and U is the eigenvector of equation (15). These eigenvectors are arranged based on the highest eigenvalues, and the eigenvector whose eigenvalues are zero or close to zero is eliminated.

$$K=P\Gamma P^T \quad (14)$$

$$(\Gamma P^T(B-W)P\Gamma)U=\lambda(\Gamma P^TWP\Gamma)U \quad (15)$$

Then, the transformation equation will be:

$$y=A^T\begin{bmatrix}\kappa(x_1,z)\\ \vdots\\ \kappa(x_N,z)\end{bmatrix} \quad (16)$$

As a result of Equation (1), the KNWFE algorithm assigns greater weights to the samples close to the center of the class, in contrary to the boundary samples. Meanwhile, as it is illustrated in Fig. 1 the boundary samples are more determinant than samples close to the center of the class. Furthermore, this method does not differentiate between classes that are close to each other, and far apart in the production of the between-class scattering matrix. Whiles, it can have a significant effect on determining the final weight of the sample. The following is a description of the proposed method for solving the problems of this algorithm.

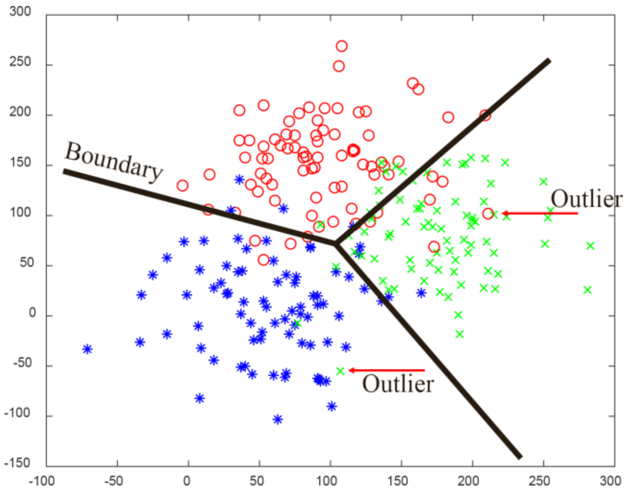


Fig. 1. The importance of boundary s in SVM classification

3- Optimized KNWFE

To solve the first problem of assigning more weight to data close to the center of the class, we use a function with Rayleigh distribution. In such a way, we pass the scattering matrix elements through this function and give higher weights to the samples near the class boundary. We then discard twenty percent of the samples in each class to prevent high weight assignment to outliers. The proposed formulations are also designed so that they take into account the distance of classes.

We define the weights $\gamma_l^{(i,j)}$ corresponding to $\lambda_l^{(i,j)}$ and another matrix $\gamma_l^{un\,(i,j)}$ so that $\gamma_l^{un\,(i,j)}$ are unnormalized weights of $\gamma_l^{(i,j)}$, and then we will have:

$$\gamma_l^{(i,j)} = \frac{\left[K_{ll}^{(i,i)} + (W^{(i,j)}K^{(j,j)}W^{(i,j)^T})_{ll} - 2(K^{(i,j)}W^{(i,j)^T})_{ll}\right]^{-1/2}}{\sum_{t=1}^{N_i}\left[K_{tt}^{(i,i)} + (W^{(i,j)}K^{(j,j)}W^{(i,j)^T})_{tt} - 2(K^{(i,j)}W^{(i,j)^T})_{tt}\right]^{-1/2}} \quad (17)$$

where i and j are indices of i^{th} and j^{th} classes, l is the index of the datum, and N_i is the total number of data in the class.

$$\gamma_l^{un\,(i,j)} = \Big[K_{ll}^{(i,i)} + (W^{(i,j)}K^{(j,j)}W^{(i,j)^T})_{ll} - 2(K^{(i,j)}W^{(i,j)^T})_{ll}\Big]^{-1/2} \quad (18)$$

First, it is needed to modify $\gamma_l^{(i,j)}$, which is the membership degree of each data in its class, as follows. In this case, the more the data is away from the center of the class (boundary data), it will gain more weight.

$$\gamma_l^{(i,i)} = \frac{\max\left(\gamma_l^{(i,i)}\right) - \gamma_l^{(i,i)}}{\sum_{t=1}^{N_i}\left\{\max\left(\gamma_t^{(i,i)}\right) - \gamma_t^{(i,i)}\right\}} \quad (19)$$

The following equation (similar to the Rayleigh distribution function) is used to weaken the effect of distorted and noisy data and remove them from the data set. In this case, we must apply this relation to the entire matrix of $\gamma_l^{(i,i)}$.

$$\gamma_l^{(i,j)} = \frac{\gamma_l^{(i,j)}}{0.606\sigma} e^{-\frac{\gamma_l^{(i,j)^2}}{2\sigma^2}} \quad (20)$$

The value of σ is the percentage of noisy data deletion. We chose this so that to consider 20% of the data of each class as offset data. That is, due to the above relationship, 20 percent of the data in a class will be weighed less and will be considered as noisy data and will not be known as boundary data. We apply the relation to the total weight of $\gamma_l^{un\,(i,j)}$. Dividing by 0.606, we normalized those weights:

$$\gamma_l^{un\,(i,j)} = \frac{\gamma_l^{un\,(i,j)}}{0.606} e^{-\frac{\gamma_l^{un\,(i,j)^2}}{2\sigma^2}} \quad (21)$$

The final weights of the data for the within-class and between-class values will be as follows:

$$\lambda_l^{(i,i)} = 0.5(\gamma_l^{(i,i)} + \max_{\forall\, i\neq j} \gamma_l^{(i,j)})\,(\max_{\forall\, i\neq j} \gamma_l^{un(i,j)})^r \quad (22)$$

$r \in [0, \infty)$

where the r parameter increases the effect of non-normalized weights on the total weights.

$$\lambda_l^{(i,j)} = 0.5(\gamma_l^{(i,i)} + \max_{\forall\, i\neq j} \gamma_l^{(i,j)})(\gamma_l^{un(i,j)})^r \quad (23)$$

$r \in [0, \infty)$

We then normalize equations (22) and (23) and obtain equations (24) and (25). Multiplying the two terms, we can change the weight between the terms, by raising one of the terms to the power of r. One may change the power r to increase the effect of $\gamma_l^{un\,(i,j)}$. Thus, the weights $\lambda_l^{(i,i)}$ and $\lambda_l^{(i,j)}$ are replaced in the original algorithm and improve the results. The next steps are similar to the original algorithm to obtain the conversion matrix.

$$\frac{\lambda_l^{(i,i)}}{\sum_l \sum_i \lambda_l^{(i,i)}} \rightarrow \lambda_l^{(i,i)} \quad (24)$$

$$\frac{\lambda_l^{(i,j)}}{\sum_l \sum_i \sum_{j\neq i} \lambda_l^{(i,j)}} \rightarrow \lambda_l^{(i,j)} \quad (25)$$

Superpixel segmentation algorithms were used along with the proposed kernel to increase the efficiency of the classification system [27]. This algorithm segments the HSI into a large number of superpixels. A superpixel consists of a combination of many contiguous pixels that have similar properties. Due to a large number of HSI bands, direct segmentation is not possible. Hence, we reduce the dimensionality using the proposed combined non-parametric kernel (CNPK) and classify it using SVM.

4- Experiments

We used three sets of HSI to evaluate the effectiveness of the proposed OKNWFE. The first set is taken from a forest-agricultural area in the northeast of the Indiana state, using the AVIRIS sensor in 1992. This image has a 220 band and 145×145 pixels. The dataset has 16 different classes and 10366 samples. Due to the absorption of radiations by the atmosphere, some of the bands are highly noisy and do not contain reliable information. Therefore, we reject the 30 noisy bands to improve the classification. The secondary data belongs to an area at Pavia University. This image is of size 610×340 and the high resolution of 1.3 meters per pixel in each band. The remaining number of channels after removing the noisy bands is 103, with a spectral range of 0.43 to 0.86 micrometers. This data includes nine different classes which are: Asphalt, meadows, gravel, trees, metal sheets, bare soil, bitumen, brick blocks, and shadows [3]. The third data belongs to the Pavia urban area, which is a 115 dimensional and 1096×715 pixel image. Removing the noisy bands, 102 bands remained for the image. This data includes nine different classes, which are water, trees, asphalt, brick blocks, bitumen, tiles, shadows, meadows, and soils.

Table I. The best-tuned used for OKNWFE and KNWFE for the AVIRIS dataset

Method	***OKNWFE***	***KNWFE***
Kernel type	Gaussian	Gaussian
Value of σ	6.6	0.5
SVM kernel type	Gaussian	Gaussian
SVM kernel width	20	20
Value of C	520000	500000
Number of k in k-NN	7	7
Distance type	Minkowski	Minkowski
Order	3	3

Table II. The best-tuned parameters for the optimized KNWFE and the KNWFE for the Pavia University Dataset

Method	***OKNWFE***	***KNWFE***
Kernel type	Gaussian	Gaussian
Value of σ	2000	220
SVM kernel type	Gaussian	Gaussian
SVM kernel width	20	20
Value of C	10000	10000
Number of k in k-NN	7	7
Distance type	Minkowski	Minkowski
Order	3	3

4-1- Parameter Setting

As in [26], we used 8 of 16 classes in AVIRIS data to evaluate the baseline and the proposed method. The simulation time for the AVIRIS data and the KNWFE method was 43 minutes, and for the OKNWFE method it was 57 minutes; for the data of the Pavia University and the KNWFE 82 minutes; for the 108-minute OKNWFE method, for the Pavia urban area data and the KNWFE method, 104 minutes, for the method OKNWFE is 130 minutes. The experiments were carried out using Core i5 3210M CPU and 6GB of RAM under the plate of MATLAB. Selected classes are, Corn-no till, Corn-min till, grass, Hay-windrowed, Soybeans-notill, Soybeans-min till, Soybeans-clean till and Woods, which are labeled class1 through class8 respectively. One thousand samples were used for each class of PAVIA urban area data. For each of the three datasets of each class, 300 samples were used to obtain the transformation matrix using the algorithm, and 350 samples were used to learn the SVM classifier. Since training data are randomly selected to train the classifier, 5-fold cross-validation is used to improve predictive performance. The Gaussian kernel sigma value variations are set at {0.02, 0.2, 2, 20, ..., 2000} for the OKNWFE method and {0.02, 0.2, 2, ..., 220} for KNWFE. The range of variations of the Gaussian kernel sigma value of the SVM class for both OKNWFE and KNWFE methods is {0.2, 2, 20, ..., 200}. Also, the range of variations of C value for both OKNWFE and KNWFE is {0.1, 1, 10, 100, 10^3, …, 10^6}. The number of neighbors in the k-NN is also {1, 3, 5, 7, 9, 11}, and the range of Minkowski distance order changes is {2, 3, 4, 5}. We have randomly test values of the σ of the Gaussian kernel for both the proposed OKNWFE and KNWFE. In the classification experiments, the best empirical values of σ are used (Table I &Table II).

Table III. The best-tuned parameters for the optimized KNWFE and KNWFE for the Pavia urban area data set

Method	***OKNWFE***	***KNWFE***
Kernel type	Gaussian	Gaussian
Value of σ	1300	2
SVM kernel type	Gaussian	Gaussian
SVM kernel width	20	20
Value of C	10000	10000
Number of k in k-NN	7	9
Distance type	Minkowski	Minkowski
Order	3	3

Table IV. The results obtained from the optimized KNWFE simulation and KNWFE for AVIRIS data

Criterions	***Overall Accuracy (%)***		***Average accuracy (%)***		***Kappa coefficient***	
Classifier	SVM	k-NN	SVM	k-NN	SVM	k-NN
OKNWFE	87.04	82.15	90.36	86.58	0.8462	0.7889
KNWFE	79.49	76.08	81.57	81.36	0.7569	0.7184

Table V. The results obtained from the optimized KNWFE simulation and KNWFE for data from the University of Pavia

Criterions	***Overall Accuracy (%)***		***Average accuracy (%)***		***Kappa coefficient***	
Classifier	SVM	k-NN	SVM	k-NN	SVM	k-NN
OKNWFE	88.73	81.10	91.86	87.26	0.8542	0.7585
KNWFE	83.88	71.08	88.45	78.12	0.7923	0.6254

Table I shows the empirically determined parameters for KNWFE and the proposed OKNWFE methods for the AVIRIS dataset. Conducting experiments on the AVIRIS dataset, it is empirically determined that the optimum value of σ for the Gaussian KNWFE is 0.5, while it is 6.6 for the Gaussian OKNWFE, and the value of *C* is 500,000 and 520,000 for KNWFE and OKNWFE respectively. The rest of the parameters are the same for both methods.

Table VI. The classification accuracy of each class of AVIRIS data using the SVM for both methods (%)

Class	1	2	3	4
OKNWFE	83.12	88.12	95.97	100
KNWFE	71.96	72.06	72.63	98.97
Class	5	6	7	8
OKNWFE	90.59	78.11	93.15	93.81
KNWFE	89.56	71.88	81.59	93.89

Table VII. The classification accuracy of each class of AVIRIS data using the k-NN for both methods (%)

Class	1	2	3	4
OKNWFE	73.01	81.53	95.77	99.59
KNWFE	59.55	73.86	94.56	99.79
Class	5	6	7	8
OKNWFE	89.87	72.24	90.71	89.95
KNWFE	85.95	66.73	83.87	86.55

Table VIII. The confusion matrix of AVIRIS data classification using the OKNWFE and SVM classifier

Class	1	2	3	4	5	6	7	8
1	1192	14	3	1	71	97	55	1
2	16	735	0	0	6	32	45	0
3	1	0	447	2	1	1	2	13
4	0	0	0	489	0	0	0	0
5	21	6	5	0	877	39	20	0
6	181	122	12	5	123	1928	97	0
7	14	12	3	0	6	7	572	0
8	0	0	74	6	0	0	0	1214

Table IX. The confusion matrix of AVIRIS data classification using the KNWFE and SVM classifier

Class	1	2	3	4	5	6	7	8
1	1032	79	4	0	114	105	100	0
2	52	601	2	1	11	86	81	0
3	0	0	361	1	0	0	2	133
4	0	0	4	484	0	0	0	1
5	36	4	6	0	867	28	27	0
6	237	135	20	4	177	1774	116	5
7	36	29	6	0	16	24	501	2
8	0	0	79	0	0	0	0	1215

Table III depicts that we empirically choose the value of 1300 and 2 for σ of OKNWFE and KNWFE, respectively. We also determine the number of K in KNN as 6 for OKNWFE and 9 for KNWFE. Table VI show the simulation results of both KNWFE and OKNWFE on all the three datasets. Table VI and Table VII how the SVM and k-NN classification accuracy on the AVIRIS data.

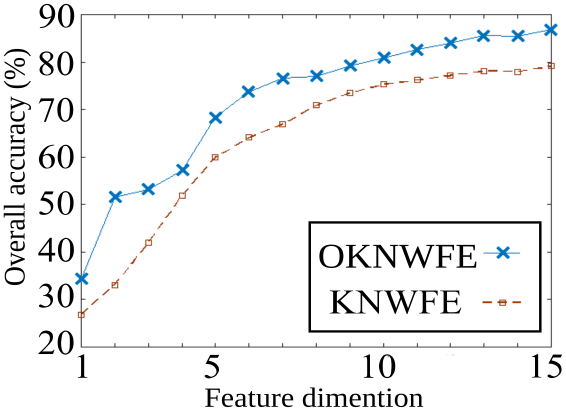


Fig. 2. Hugh diagram for OKNWFE and KNWFE using

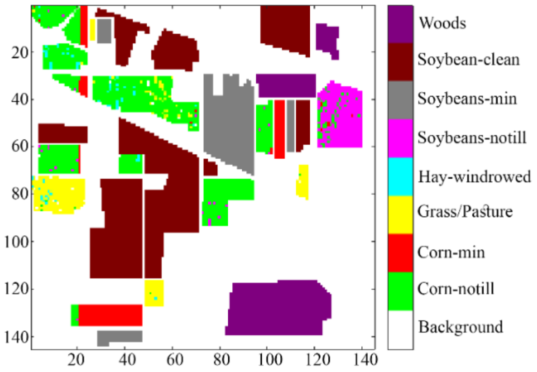


Fig. 3. OKNWFE classification map using SVM classification for the AVIRIS data

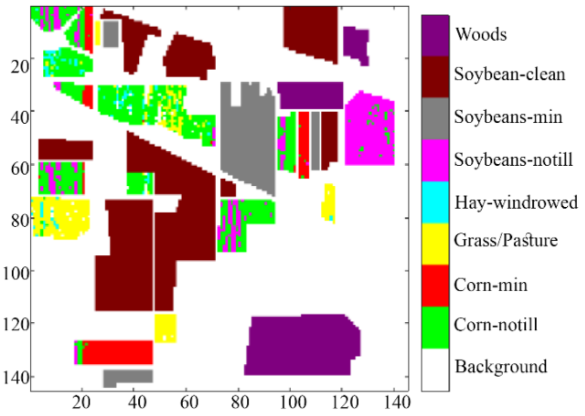


Fig. 4. Classification of the KNWFE method map using the SVM classification for the AVIRIS data

4-2- Dispersion Map Analysis

To prevent the same results from being exceeded, we refrain from providing details of the results of other datasets. In the following, only the results of the SVM classification for the AVIRIS data are given. Table VIII and Table IX show the confusion matrices for both methods using the SVM classifier on the AVIRIS dataset. As the results of the experiments show, the proposed method with optimized data weights is superior to the KNWFE method. This improvement will come at the expense of increased computing.

The Hugh diagram is shown in Fig. 2. This figure shows the classification accuracy concerning the number of features. As it is clear from the curve, the accuracy of both the methods increases with the number of features until it goes to a so-called saturation. This also demonstrates that the OKNWFE method can achieve better performance for all the number of features.

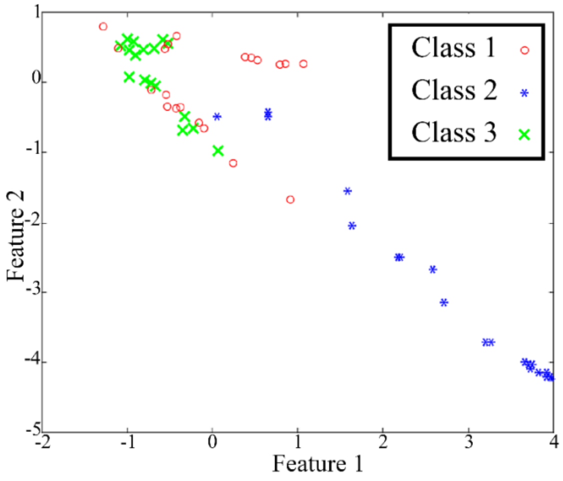


Fig. 5. Dispersion map of OKNWFE for the AVIRIS data

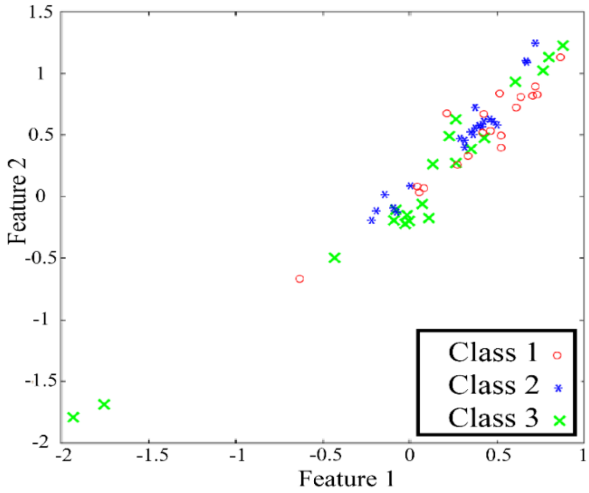


Fig. 6. Dispersion map of the KNWFE method for the AVIRIS data

Fig. 3 and Fig. 4 represent the classification map, where each class represented with a color. Spots on some classes, which are more pronounced in the corn-notill class, indicate classification errors. As the comparison of the two images clearly shows, this error in OKNWFE is far less than in KNWFE. Fig. 5 and Fig. 6 show the dispersion map for both the methods using SVM classifier on AVIRIS data. As it is clear from Fig. 2, with the increase in the number of features, the separability of classes increases. As the number of features increases, the slope of the curve decreases and eventually reaches almost zero. It is also clear that the proposed method has more classification accuracy than the KNWFE method at each step and with the same number of attributes.

4-3- Experimental Results and Analysis

In this paper, we propose a hybrid non-parametric optimized kernel method for HSI classification, comparing its performance with some of the state-of-the-art methods of HSI classification as baseline methods. These methods include: SC-MK [28], RMK [29], RPNet [30]. Table X compares the performance of the base methods with the proposed CNPK method for the PAVIA university database. This table shows the classification efficiency of each class as well as the overall performance (OA). Each method was performed ten times using randomly selected samples to ensure the generality of the results, and we entered the average accuracy for each method in the table. The results show that the proposed method is more efficient in most classes than other methods. Also, in overall performance, the proposed method shows better results than all classes.

Table X Classification performance for PAVIA university dataset

Class name	***SC-MK***	***RMK***	***RPNet***	***CNPK***
Asphalt	0.8279	**0.9821**	0.952	0.9709
Meadows	0.9083	**0.9783**	0.9663	0.9668
Gravel	0.9176	0.9588	0.8856	**0.9781**
Trees	0.9652	0.905	0.9618	**0.9676**
Metal sheets	0.9999	0.9715	0.9634	**0.9999**
Bare soil	0.9711	0.9902	0.9088	**0.9937**
Bitumen	0.9601	0.9923	0.7825	**0.9994**
Bricks	0.9063	**0.9731**	0.9306	0.9725
Shadows	0.9682	0.5602	0.8222	**0.9927**
OA	0.9361	0.9235	0.9082	**0.9824**

The same experiment is performed on the PAVIA urban database. Table XI illustrates the classification results for this database.

Table XI Classification performance for PAVIA urban area dataset

Class name	***SC-MK***	***RMK***	***RPNet***	***CNPK***
Water	0.9992	0.9739	0.9952	**1.0000**
Tree	0.9186	0.8222	0.9008	**0.9573**
Asphalt	0.9723	0.9289	0.969	**0.9755**
Blocking Bricks	0.9904	0.9771	**0.9936**	0.9924
Bitumen	**0.9978**	0.9573	0.9768	0.9735
Tiles	**0.994**	0.9667	0.9618	0.9683
Shadows	0.9684	0.9752	0.9273	**0.9873**
Meadows	0.9831	0.9262	0.948	**0.99**
Bare Soil	0.9719	0.8162	0.9765	**0.9863**
OA	0.9773	0.9271	0.961	**0.9812**

5- Conclusion and Discussion

In this paper, we propose a feature extraction method that reduces the dimensions of a hyperspectral image so that the different segments of the image are better distinguishable. The method, which is called OKNWFE, results in the improvement of HSI classification. This improvement is obtained at the cost of an increase in computation complexity. As shown in Table IV to Table VI and Fig. 2 to Fig. 6, the OKNWFE method outperforms KNWFE. The dispersion map, drawn for the first and second characteristics, shows that the OKNWFE method provides better separation than the KNWFE method, and the classification map shows that the proposed method has less error in the classification of classes had. The experimental results suggest that the proposed method, in combination with the superpixel segmentation algorithms, has superior performance to the state-of-the-art systems for HIS classification. Based on the results and the proposed method, suggestions can be made for future research. As future work, one may use this kernel on CNNs for classifying hyperspectral images. This kernel can be an activation function on CNN. Adapting this kernel to achieve an optimal CNN could be the subject of future research. The type and number of other layers besides this activation function is another subject worth exploring. Calculating optimal kernel parameters for kernel-based methods using cross-validation is very time-consuming, so studying computation reduction methods can be a useful study. Besides, one may use synthetic kernels and find combination parameters using the classical optimization method.

References

[1] H. Li, H. Zhou, L. Pan, and Q. Du, “Gabor feature-based composite kernel method for hyperspectral image classification,” vol. 54, no. 10, 2018, doi: 10.1049/el.2018.0272.

[2] D. Hong, X. Wu, P. Ghamisi, J. Chanussot, N. Yokoya, and X. X. Zhu, “Invariant Attribute Profiles: A Spatial-Frequency Joint Feature Extractor for Hyperspectral Image Classification,” IEEE Trans. Geosci. Remote Sens., pp. 1–18, 2020, doi: 10.1109/TGRS.2019.2957251.

[3] S. Suresh and S. Lal, “A Metaheuristic Framework based Automated Spatial-Spectral Graph for Land Cover Classification from Multispectral and Hyperspectral Satellite Images,” Infrared Phys. Technol., vol. 105, no. January, p. 103172, 2020, doi: 10.1016/j.infrared.2019.103172.

[4] P. Xiang et al., “Hyperspectral anomaly detection by local joint subspace process and support vector machine,” Int. J. Remote Sens., vol. 41, no. 10, pp. 3798–3819, 2020.

[5] P. Ghamisi, J. Plaza, Y. Chen, J. Li, and A. J. Plaza, “Advanced spectral classifiers for hyperspectral images: A review,” IEEE Geosci. Remote Sens. Mag., vol. 5, no. 1, pp. 8–32, 2017.

[6] L. Fang, Z. Liu, and W. Song, “Deep Hashing Neural Networks for Hyperspectral,” IEEE Geosci. Remote Sens. Lett., vol. PP, pp. 1–5, 2019, doi: 10.1109/LGRS.2019.2899823.

[7] E. M. Paoletti, M. J. Haut, J. Plaza, and A. Plaza, “Deep&Dense Convolutional Neural Network for Hyperspectral Image Classification,” Remote Sens., vol. 10, no. 9, pp. 1–21, 2018, doi: 10.3390/rs10091454.

[8] H. Lee, M. Kim, D. Jeong, S. Delwiche, K. Chao, and B.-K. Cho, “Detection of cracks on tomatoes using a hyperspectral near-infrared reflectance imaging system,” Sensors, vol. 14, no. 10, pp. 18837–18850, 2014.

[9] B.-C. Kuo and D. A. Landgrebe, “Nonparametric weighted feature extraction for classification,” IEEE Trans. Geosci. Remote Sens., vol. 42, no. 5, pp. 1096–1105, 2004.

[10] M. R. Almeida, L. P. L. Logrado, J. J. Zacca, D. N. Correa, and R. J. Poppi, “Raman hyperspectral imaging in conjunction with independent component analysis as a forensic tool for explosive analysis: The case of an ATM explosion,” Talanta, vol. 174, pp. 628–632, 2017.

[11] Z. Chen, J. Jiang, X. Jiang, X. Fang, and Z. Cai, “Spectral-spatial feature extraction of hyperspectral images based on propagation filter,” Sensors (Switzerland), vol. 18, no. 6, pp. 1–16, 2018, doi: 10.3390/s18061978.

[12] J. Jiang, J. Ma, C. Chen, Z. Wang, Z. Cai, and L. Wang, “SuperPCA: A Superpixelwise PCA Approach for

Unsupervised Feature Extraction of Hyperspectral Imagery," IEEE Trans. Geosci. Remote Sens., vol. 56, no. 8, pp. 4581–4593, Aug. 2018, doi: 10.1109/TGRS.2018.2828029.

[13] H. Su, S. Member, B. Zhao, Q. Du, P. Du, and S. Member, "With Local Correlation Features for Hyperspectral Image Classification," IEEE Trans. Geosci. Remote Sens., vol. PP, pp. 1–12, 2018, doi: 10.1109/TGRS.2018.2866190.

[14] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," IEEE Trans. Geosci. Remote Sens., vol. 43, no. 6, pp. 1351–1362, Jun. 2005, doi: 10.1109/TGRS.2005.846154.

[15] M. Khodadadzadeh, P. Ghamisi, C. Contreras, and R. Gloaguen, "Subspace Multinomial Logistic Regression Ensemble for Classification of Hyperspectral Images," in IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, Jul. 2018, pp. 5740–5743, doi: 10.1109/IGARSS.2018.8519404.

[16] S. Song, H. Zhou, J. Zhou, K. Qian, K. Cheng, and Z. Zhang, "Hyperspectral anomaly detection based on anomalous component extraction framework," Infrared Phys. Technol., vol. 96, pp. 340–350, 2019, doi: 10.1016/j.infrared.2018.12.008.

[17] E. Blanzieri and F. Melgani, "Nearest Neighbor Classification of Remote Sensing Images With the Maximal Margin Principle," IEEE Trans. Geosci. Remote Sens., vol. 46, no. 6, pp. 1804–1811, Jun. 2008, doi: 10.1109/TGRS.2008.916090.

[18] D. Tuia and G. Camps-Valls, "Semisupervised Remote Sensing Image Classification With Cluster Kernels," IEEE Geosci. Remote Sens. Lett., vol. 6, no. 2, pp. 224–228, Apr. 2009, doi: 10.1109/LGRS.2008.2010275.

[19] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral Image Classification via.pdf," vol. 51, no. 1, pp. 217–231, 2013.

[20] X. Weng, W. Lei, and X. Ren, "Kernel sparse representation for hyperspectral unmixing based on high mutual coherence spectral library," Int. J. Remote Sens., vol. 41, no. 4, pp. 1286–1301, 2020, doi: 10.1080/01431161.2019.1666215.

[21] Y. Xu, Z. Wu, J. Chanussot, and Z. Wei, "Nonlocal Patch Tensor Sparse Representation for Hyperspectral Image Super-Resolution," IEEE Trans. Image Process., vol. 28, no. 6, pp. 3034–3047, 2019, doi: 10.1109/TIP.2019.2893530.

[22] G. Cheng, Z. Li, J. Han, X. Yao, and L. Guo, "Exploring Hierarchical Convolutional Features for Hyperspectral Image Classification," IEEE Trans. Geosci. Remote Sens., vol. PP, pp. 1–11, 2018, doi: 10.1109/TGRS.2018.2841823.

[23] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "ISPRS Journal of Photogrammetry and Remote Sensing A new deep convolutional neural network for fast hyperspectral image classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 120–147, 2018, doi: 10.1016/j.isprsjprs.2017.11.021.

[24] B. Pan, Z. Shi, and X. Xu, "ISPRS Journal of Photogrammetry and Remote Sensing MugNet : Deep learning for hyperspectral image classification using limited samples," *ISPRS J. Photogramm. Remote Sens.*, 2017, doi: 10.1016/j.isprsjprs.2017.11.003.

[25] O. Okwuashi and C. E. Ndehedehe, "Deep support vector machine for hyperspectral image classification," Pattern Recognit., vol. 103, pp. 2–25, 2020, doi: 10.1016/j.patcog.2020.107298.

[26] B. C. Kuo, C. H. Li, and J. M. Yang, "Kernel nonparametric weighted feature extraction for hyperspectral image classification," IEEE Trans. Geosci. Remote Sens., vol. 47, no. 4, pp. 1139–1155, 2009, doi: 10.1109/TGRS.2008.2008308.

[27] L. Sun, C. Ma, Y. Chen, H. J. Shim, Z. Wu, and B. Jeon, "Adjacent superpixel-based multiscale spatial-spectral kernel for hyperspectral classification," IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens., vol. 12, no. 6, pp. 1905–1919, 2019.

[28] T. Zhan, L. Sun, Y. Xu, G. Yang, Y. Zhang, and Z. Wu, "Hyperspectral classification via superpixel kernel learning-based low rank representation," Remote Sens., vol. 10, no. 10, p. 1639, 2018.

[29] J. Liu, Z. Wu, Z. Xiao, and J. Yang, "Region-based relaxed multiple kernel collaborative representation for hyperspectral image classification," IEEE Access, vol. 5, pp. 20921–20933, 2017.

[30] Y. Xu, B. Du, F. Zhang, and L. Zhang, "Hyperspectral image classification via a random patches network," ISPRS J. Photogramm. Remote Sens., vol. 142, pp. 344–357, 2018.

Journal of Information Systems and Telecommunication
Vol.10, No.2, April-June 2022, 120-131

http://jist.acecr.org
ISSN 2322-1437 / ESSN:2345-2773

Self-Organization Map (SOM) Algorithm for DDoS Attack Detection in Distributed Software Defined Network (D-SDN)

Mohsen Rafiee[1], Alireza Shirmarz[2*]

[1].Department of Computer Engineering, University of Mazandaran, Mazandaran, Iran
[2].Department of Computer & Electronic Engineering, Ale-Taha University, Tehran, Iran

Received: 05 Dec 2020/ Revised: 15 Sep 2021/ Accepted: 21 Nov 2021
DOI:

Abstract

The extend of the internet across the world has increased cyber-attacks and threats. One of the most significant threats includes denial-of-service (DoS) which causes the server or network not to be able to serve. This attack can be done by distributed nodes in the network as if the nodes collaborated. This attack is called distributed denial-of-service (DDoS). There is offered a novel architecture for the future networks to make them more agile, programmable and flexible. This architecture is called software defined network (SDN) that the main idea is data and control network flows separation. This architecture allows the network administrator to resist DDoS attacks in the centralized controller. The main issue is to detect DDoS flows in the controller. In this paper, the Self-Organizing Map (SOM) method and Learning Vector Quantization (LVQ) are used for DDoS attack detection in SDN with distributed architecture in the control layer. To evaluate the proposed model, we use a labelled data set to prove the proposed model that has improved the DDoS attack flow detection by 99.56%. This research can be used by the researchers working on SDN-based DDoS attack detection improvement.

Keywords: Software Defined Network (SDN); Distributed Controller; Distributed Denial-of-Service (DDoS); Self-Organizing Map (SOM); Learning Vector Quantization (LVQ).

1- Introduction

Internet extend has been raised across the world sharply, so internet technology usage rate among business and the social activities went up. The complexity of the traditional network architecture on the internet exposes the network specialist to a situation that makes the configuration and network control impossible, so the scientists proposed a new architecture called software defined network (SDN) to be used for future networks [1]. The SDN includes three layers, application, control, and data. There are various defined tasks for each layer and this structure makes the network much more programmable, flexible, and manageable [2][3]. The SDN, in addition to three layers, has three APIs (Northbound, Southbound, and East-West) to connect the layers and scale the controller with controllers' communication capability [2][4][5][6][7] which Fig.1 shows the layers and APIs, briefly. The data plane is composed of FEs which are simple forwarding elements. The control plane has the main role of decision-making in SDN. The controller can be a physical centralized or conceptual centralized controller. The conceptual centralized controller is composed of some controllers which are related together with east-west APIs. The data plane and control plane are connected by southbound API. The application layer is based on network applications and is connected with the control plane with northbound APIs.

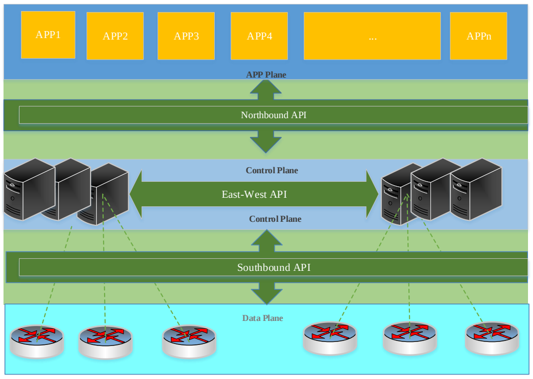


Fig. 1 SDN Architecture

✉ **Alireza Shirmarz**
a.shirmarz@aut.ac.ir

1-1- Problem Statement

One of the significant attacks on the internet is DDoS which has laid in the central attention of the published recent papers. The papers have proposed approaches to mitigate DDoS attacks with the use of machine learning (ML) or statistical methods. The DDoS attacks have been categorized in [8]. The DDoS is a malicious effort to disrupt the normal network flows that are done with fake traffic generating as if the service cannot be provided by the server or the network. The denial of service takes place because the computing resources are busy with fake traffic. According to Cisco's annual internet report which has been published in 2020, the number of DDoS attacks will double to 15.4 million by 2023 globally [9] as shown in Fig. 2.

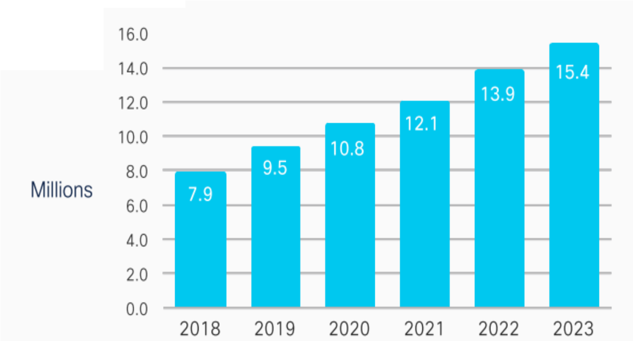


Fig. 2 The prediction of DDoS Attack growth by 2023 [9]

The limitation in memory and processing resources that exist in smart devices has caused the networks to become more susceptible to large-scale DDoS attacks. Internet of things (IoT) expands the use of smart devices which makes this issue more important [10]. For instance, the DDoS attacks which happened in well-known companies and organizations like CNN, Netflix, Twitter caused a denial of service in 2006 [11]. These reasons show the importance of DDoS attacks and their effects on IT. The taxonomy of the DDoS attacks' types are organized in a tree which is presented in Fig.3.

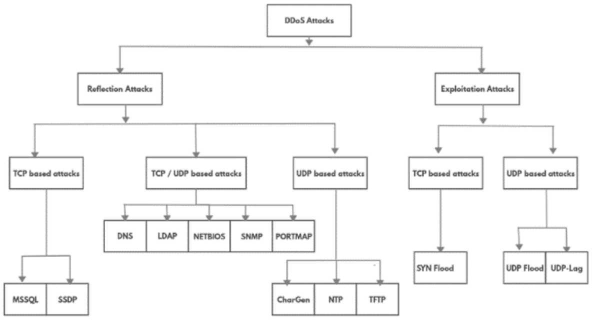


Fig. 3 DDoS Attacks Taxonomy [8]

One of the SDN traits is the (physical or conceptual) centeralized controller that has a global view of the network to make the optimized decision; therefore, it can make the network more secure from different attacks in comparison with the traditional networks. One of the SDN architecture drawbacks is a single point of failure of the controller; hence, the controller has been posed threats by the attackers that can make the controller denial of service. Due to this defect, distributed controllers architecture has been proposed which is addressed in this paper. This distributed controllers architecture has solved the single point of failure and mitigate the DDoS attack on the centeralized controller.

The first step to encounter DDoS is to detect the attack with a proper solution. Firewalling, intrusion detection system (IDS), and intrusion prevention system (IPS) can be developed and deployed in the application layer. The solutions in the application layer are facing problems like:

- Application layer workload
- Less smart behaviour
- Expert administrator
- Less Agility
- Interference in the defined rules
- The symmetry of the rules

1-2- Proposed Approach

In this paper, we propose the DDoS attack detection module in the controller with distributed architecture. The proposed solution is based on a machine learning method named self-organizing map (SOM) which is used for smart DDoS attack detection. There is a wide diversity in the flow patterns for DDoS attacks; therefore, it is needed to extract flows pattern automatically. The SOM is powerful in pattern extraction from real data. This pattern extraction lets us detect the new patterns for a DDoS attack. This algorithm can be used for classification and IDS as anomaly detection. The model is extracted from the dataset which is CICDDoS2019 and was obtained from the Canadian Institute for Cybersecurity, University of New Brunswick, Canada. This comprehensive dataset consists of 50063112 instances with 76 features along with 13 class labels to predict DDoS attacks [12]. The proposed approach will be compared with the related works and show that our model works with more accuracy. The proposed clustering algorithm labels the dataset and this method is compared with the labelled dataset. In this paper, a distributed architecture is used for the proposed model. The proposed model is simulated in Mininet and shows that the proposed model improves the DDoS attacks detection.

In this paper, the novelties that are proposed are:

- The Self-Organizing Map (SOM) method and Learning Vector Quantization (LVQ) are used for DDoS attack detection

- The DDoS attack detection model in distributed SDN controllers
- DDoS attack detection improvement in SDN

In the following section, the background of the concept will be discussed.

2- Background

2-1- SDN Functionality

In traditional networks, all switching and routing decisions are done in the switches, routers, firewalls and other equipment while these decisions are done in the centralized controller in SDN architecture. In SDN, there is only equipment instead of switches, routers, firewalls which is responsible for forwarding called forwarding element (FE). The other spec of SDN is the flow-based decision instead of the packet-based one. There is a definition for flow which is a sequence of packets with a common source and destination which is determined with five tuples (source IP, destination IP, source port number, destination port number and transport protocol). The packet header is extracted in each FE then the existence of the rule inside the flow table for the packet is examined. If the packet is the first packet of a flow and doesn't exist in the flow table, FE forwards the packet-in message including the packet header and payload to the controller. The controller makes a proper decision for each flow and exports the rule to each FE to fill the flow table. If the packet exists in the flow table, FE forwards the packet based on the action defined in the flow table. The controller decision for each flow has been made based on the network policy planned by the network administrators. The flow action is exported with the flow-mode message [13].

The rule of flow is embedded in the flow table which has been placed in FEs. The FE accomplish according to the action which is put in the flow table. The flow table is flushed in two cases, soft time-out, and hard time-out. The soft timeout depends on the idle flow entry which exists in the flow-table, and hard time-out refers to the period of the time in which the flow-table should be made empty by force. This mechanism makes memory space more efficient.

2-2- Self-Organizing Map Algorithm

The SOM algorithm has been proposed in 1982 by Kohonen [14]. This is an unsupervised learning algorithm that learns the patterns from complex datasets and clusters the data with noise. It is a neural network-based dimensionality reduction algorithm generally used to represent a high-dimensional dataset as two dimensional discretized pattern. Dimensionality reduction is performed while retaining the topology of data present in the original feature space. The clustering method is a k-means clustering performed on the mapping generated by SOM. As the first step, an artificial neural network is trained to generate a low-dimensional discretized representation of the data in the original feature space while preserving the topological properties; this is achieved through competitive learning. In SOM, the vectors that are close in the high-dimensional space also end up being mapped to SOM nodes that are close in low-dimensional space. K-means can be considered a simplified case of SOM, where the nodes (centroids) are independent of each other. K-means is highly sensitive to the initial positions of the centroids, and it is not suitable for a high-dimensional dataset. The two-stage procedure for clustering adopted in this study first uses SOM to produce the low-dimensional prototypes (abstractions) that are then clustered in the second stage using k-means. This two-step clustering method reduces the computational time and improves the efficiency of K-means clustering. Even with a relatively small number of samples, many clustering algorithms especially hierarchical ones become intractably heavy. Another benefit of the two-step clustering method is noise reduction. The prototypes constructed by SOM are local averages of the data; therefore, less sensitive to random variations than the original data. The weights of SOM were randomly initialized. During training, the weight vectors are updated based on the similarity between the weight vectors and input vectors which results in moving the SOM neurons/nodes closer to certain dense regions of the original data. The similarity between data points and SOM nodes during the weight update is evaluated based on Euclidean distance [15] as shown in Fig. 4. The main steps which are required in SOM are:

1) Train step: the neurons' network weights are determined with trained sets.
2) Map step: the winner neurons are chosen and clustered automatically.

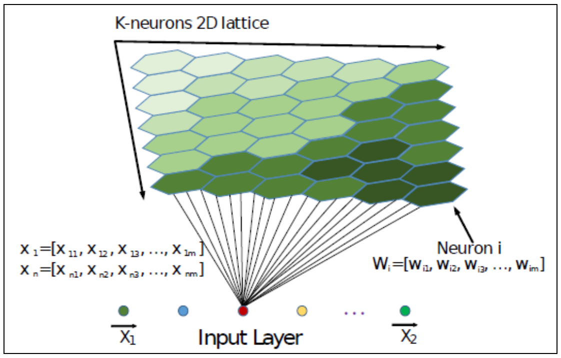

K-neurons 2D lattice
$x_1=[x_{11}, x_{12}, x_{13}, \ldots, x_{1m}]$
$x_n=[x_{n1}, x_{n2}, x_{n3}, \ldots, x_{nm}]$
Neuron i
$W_i=[w_{i1}, w_{i2}, w_{i3}, \ldots, w_{im}]$
$\overrightarrow{X_1}$
Input Layer
$\overrightarrow{X_2}$

Fig. 4 The Self-organizing map [15]

The SOM algorithm defines some variables which are [16]:

- N: The number of training instances. These instances are shown in a set [$\vec{x}_1$, $\vec{x}_2$, …, $\vec{x}_n$]
- S: The number of neurons has been mapped. These neurons are presented by the vector with m dimensions.
- The lattice radius is defined as follows:

$$R = \frac{\max(Map\ Width,\ MapHeight)}{2} \quad (1)$$

- λ is fixed and calculated as:

$$\lambda = \frac{N}{\log R} \quad (2)$$

- σ(t) is the winner neurons radius as shown in Fig. 5. This radius is calculated by the dependent period like t.

$$\sigma(t) = R \times \exp\left(-\frac{t^2}{\lambda}\right) \qquad t = 1, \ldots, n \quad (3)$$

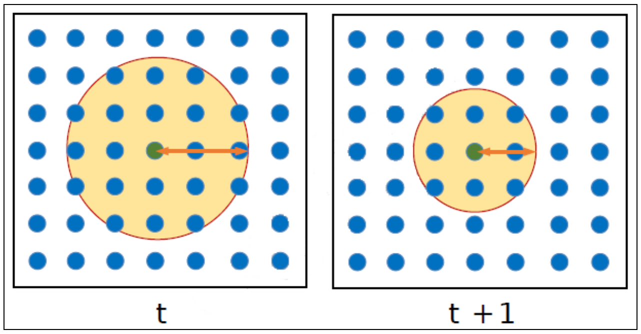


Fig. 5 The radius degradation in the map [14]

The SOM is computed in three phases:

1) **Competition Phase:** In this stage, neurons compete to choose the center of each cluster. For this purpose, the neurons are first given the initial value. The dimensions of these values are the same as the input data.

$$W_i = [\vec{W}_{i1}, \vec{W}_{i2}, \ldots, \vec{W}_{im}] \quad 1 \le i \le S \quad (4)$$

The cluster with the smallest distance to the input vector x_k = [x_{k1}, x_{k2}, …, x_{km}] wins the competition. There are several methods for determining the distance between neurons and the input vector. In this article, the Euclidean distance is used and can be expressed as follows:

$$D_{ist} = \sqrt{\sum_{j=0}^{m} (x_{kj} - W_{ij})^2} \quad (5)$$

2) **Collaboration Phase:** In this stage, the effect of winning i neurons on the learning of neighbouring neurons when applying input x is calculated. The greater the distance between neighbouring neurons and the winning neuron, the less effective it is. Experience has shown that it is best to consider a large neighbourhood first to include dead neurons. The magnitude of this effect is calculated from the following equation:

$$\Theta(t) = \exp\left(-\frac{D_{ist}^{\ 2}}{2\sigma(t)^2}\right) \quad (6)$$

3) **Adaptation Phase:** In this stage, according to the equation under the weight of the neurons, they are optimized for the next repetitions so that over time, the whole map converges towards the input vector:

$$W(t+1) = W(t) + L(t) \times \Theta(t) \times (x_k(t) - W(t)) \quad (7)$$

The variable L(t) is the learning rate that decreases over time. It is calculated as:

$$L(t) = L_0 \times \exp\left(-\frac{t}{\lambda}\right) \quad (8)$$

Collaboration and adaptation are repeated to enter the mapping stage to complete network learning.

2-3- DDoS Attacks

To detect DDoS attacks, it is needed to know the characteristics of this attack. Generally, DDoS attackers disable the targeted victim by anomaly fake network traffic generating. There are different categories that have been proposed for DDoS attack types. In this paper, Figure 3 shows a classification of DDoS and includes two major general modes. The first DDoS attack model is based on the occupied victim system's bandwidth by large packets as if the system cannot service like the DNS service attack. The second model that can be done as the DDoS attack is to disrupt the main resources of the victim system such as memory and processor by sending unusual and abnormal packets such as syn-flood.

3- Related Works

In this section, we review SDN security researches related to DDoS attack detection mechanisms and address approaches similar to the proposed method. DDoS attack detection in SDN can be categorized into two methods which are based on statistical analysis methods and Machine Learning (ML) methods.

In [17]–[20] researches, statistical analysis is used to detect DDoS attacks in SDN. The entropy method is one of the most widely used statistical analysis methods to detect DDoS attacks. Entropy is a parameter to measure randomness. That is, it determines the probability of an event occurring according to the total number of events. The higher the randomness, the higher the entropy. In [21] a threshold-based entropy method has been proposed to detect DDoS attacks in SDN. In a network, each host must receive new packets with a probability that is almost close to each other, in which case the entropy will be high. If one or more hosts receive too many packets, the randomness decreases and as a result, the entropy will decrease. According to this, a threshold being set for entropy, and the attack will be detected if the entropy value falls below this threshold. In [22] the controller periodically creates a hash table of destination addresses through the information received from the switches. The entropy of the destination addresses is then calculated by the phi-entropy method. A DDoS attack is detected if the entropy value is below the threshold for more than five consecutive windows.

The growth of Machine Learning knowledge makes many researchers use it in DDoS attack detection. One of the important and key steps in the use of Machine Learning is to select the appropriate features for learning the algorithm. In paper [23] It uses the Dynamic MLP method to select the optimal features. Polat et al It has used Support Vector Machine (SVM), Naive Bayes (NB), Artificial Neural Network (ANN), and K-Nearest Neighbors (KNN) machine learning methods to detect DDoS attack in SDN and has reached 98.3% in diagnosis accuracy with KNN method. Phan et al in [24] have proposed a Distributed SOM DSOM method to detect DDoS attacks in SDN. In this paper, the control plane architecture is Centralized with a POX controller. The proposed method also uses multiple self-organizing maps integrated with OpenFlow switches instead of a self-organizing one. Each DSOM in each switch processes the incoming traffic; hence, the processing load on the controllers will be divided between the switches. There is also a DSOM component in the application layer that is responsible for managing the performance of DSOM on switches. Due to the fact that in this method, the attack detection point is located in the switches and in the data layer, it is necessary to check all the packets passing through the switch by SOM, which consumes a lot of processing time and time. Braga et al in [25] introduce the Lightweight DDoS Flooding Attack method that detects a DDoS attack based on tracking suspicious input flows using self-organizing mapping. The control plane architecture is centralized with a NOX controller. The SOM features are extracted from the incoming traffic. Each instance then enters a SOM map to determine whether the incoming traffic flow is normal or malicious. This method includes three components Flow Collector, Feature Extractor and Classifier. The papers which are [26], [27] include other works that have used machine learning techniques to detect DDoS attacks in SDN.

T. Nam and et al have used SDN and proposed SOM and K-NN clustering to detect DDoS attacks in [28]. They have worked on DDoS attack detection and examined it with different k. They could find out the best accuracy for DDoS attacks with k=3. They could reach the accuracy %99.05 which is noticeable; therefore, we will compare our result with this paper.

The use of distributed controller architecture in the control layer and the application of distinctive features in the attack detection stage are the most important differences between our method and the above-mentioned works, which are discussed in this article.

4- The Proposed Scheme for DDoS Attack Detection in SDN

As mentioned in the proposed method, the components of DDoS attack detection are managed by controllers. The attack detection point will be at the level of the control layer of software-based networks. To achieve this goal, it is necessary to go through the four main steps shown in Fig.6 which in the proposed method will focus on the first and second steps:

1) Data Gathering: At this stage, appropriate statistical information should be obtained from network traffic so that normal traffic can be distinguished from attack traffic.
2) Attack Detection: A method should be implemented based on which the occurrence of the attack can be detected by entering the information collected in the previous step in the output. In this dissertation, machine learning in the proposed method is used.
3) Decision making: In this stage, it is determined what decisions should be made after identifying the attack. The main point of decision is in the network controller.
4) Execution: Converts the decisions made to the input of the flows and then applies them to switches and routers, such as deleting flow entry from the flow table.

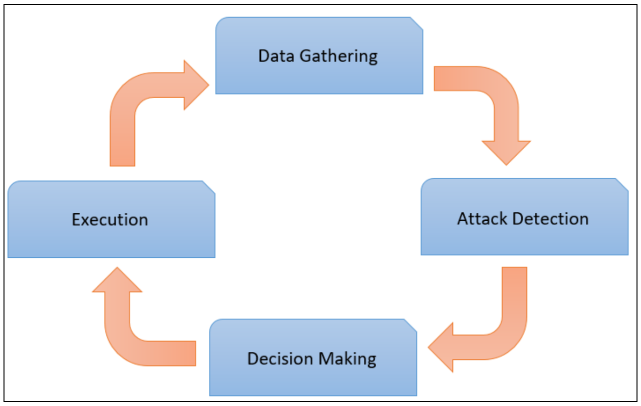


Fig. 6 Main four phases

In the following, first, the processing performed on the dataset used and how the self-organizing mapping algorithm works will be discussed. Then the attack simulation scenario is introduced along with the tools used to generate and analyze attack traffic.

4-1- Dataset Introduction

This paper uses the CICDDoS2019 dataset [8] compiled by the University of New Brunswick. This dataset includes normal traffic and the most up-to-date distributed denial-of-service attack traffic, which includes various types of attack traffic such as DNS, LDAP, SYN, etc.
To extract the feature, the CICFlowMeter two-way traffic flow generation tool was used. The output of this tool contains 76 features.

4-2- Data Pre-Processing

The Weka tool has been used to facilitate and expedite dataset processing. Weka is a Java-based machine learning tool developed at the University of Waikato in New Zealand. Weka allows users to extract useful information from the database. The heterogeneous dataset is used for classification. That is, the ratio of normal traffic to attack is very different, so-called unbalanced data. When the data is unbalanced, the performance of the detection algorithm cannot be properly evaluated. Because the neurons of the machine learning algorithm are biased towards the traffic class that has the largest number. Therefore, in order to balance the dataset, using the Weka tool, the Random Under Sampling method, which is one of the data Mining methods, has been used to eliminate the number of existing samples to reach balanced.
As mentioned, this dataset contains 76 features. One way is to use all of these features to detect an attack in a machine learning algorithm. However, due to a large number of features, it can prolong the processing time. On the other hand, the effect of all these features in identifying traffic related to a DDoS attack will not be the same. Some features will be more effective and some will be less effective. Choosing the most effective features is a major challenge in itself. As a result, the best solution is to be able to use all of the features in some way, as well as reduce the processing load by reducing the number of features. Therefore, in this paper, the feature extraction method [29] with the principle component analysis (PCA) has been used to obtain new features from the entire dataset. The method of calculating the new properties can be calculated as follows: where A_i is the main property of i, $A^{'}_i$ is the new property of i, V_{ij} is the component j of the vector i Eigenvector, n is the number of new properties obtained after PCA and m is the maximum number of main properties involved. In linear transmission are:

$$A'_i=V_{11}\times A_1+V_{12}\times A_2+\ldots+V_{im}\times A_m$$
$$=\sum_{j=1}^{m} v_{ij} \times A_j, \quad i=1,2,\ldots,n$$
(9)

To reduce the number of features, the PCA has been used and decreased the attributes from 76 to 23. Table 1 shows the obtained value after PCA processing that some are shown in the following table.

Table 1: PCA extracted features

#	PCA Extracted Features
1	-0.247Flow IAT Max - 0.246Fwd IAT Max - 0.244Idle Max - 0.241Idle Mean + …
2	-0.304Pkt Len Mean - 0.291Pkt Size Avg - 0.278Fwd Seg Size Avg + …
3	-0.235Subflow Bwd Byts - 0.235TotLen Bwd Pkts - 0.211Bwd Pkt Len Mean + …
4	-0.375Fwd Act Data Pkts - 0.353Tot Fwd Pkts - 0.353Subflow Fwd Pkts + …
5	0.476Active Mean + 0.441Active Max + 0.429Active Min + 0.306Flow IAT Min + …
6	-0.387Fwd PSH Flags - 0.387RST Flag Cnt - 0.294Flow Pkts/s - 0.293Fwd Pkts/s + …
7	-0.374Fwd PSH Flags - 0.374RST Flag Cnt - 0.333URG Flag Cnt + …
8	0.425CWE Flag Count + 0.405Down/Up Ratio - 0.372ACK Flag Cnt + …
9	-0.508Fwd IAT Min - 0.506Flow IAT Min - 0.229Flow IAT Mean + …
10	-0.464Bwd Pkt Len Min + 0.339CWE Flag Count + 0.292Init Fwd Win Byts + …
11	0.326Flow IAT Mean - 0.285Bwd IAT Max - 0.279Bwd IAT Tot + …
12	-0.715Fwd Header Len - 0.687Fwd Seg Size Min - 0.043Fwd Pkt Len Std + …

13	-0.418Init Bwd Win Byts + 0.342Bwd Pkts/s + 0.326Active Std + …
14	-0.464Active Std - 0.369Bwd IAT Mean + 0.31 Active Min + 0.304Idle Min + …
15	0.403Bwd IAT Mean + 0.318Bwd IAT Std - 0.294Fwd Pkt Len Std + …
16	0.687Bwd Header Len - 0.604SYN Flag Cnt - 0.246Bwd Pkts/s + …
17	-0.691SYN Flag Cnt - 0.658Bwd Header Len + 0.121CWE Flag Count + …
18	-0.641Bwd Pkts/s - 0.604Bwd IAT Min - 0.27Bwd Header Len + …
19	-0.643Bwd IAT Min + 0.558Bwd Pkts/s + 0.235Bwd Pkt Len Min + …
20	-0.38Init Bwd Win Byts - 0.377Bwd Pkt Len Min + 0.319Fwd Pkt Len Std + …
21	0.564Idle Std + 0.342Fwd Pkt Len Std - 0.282Active Std - 0.208Pkt Len Var + …
22	-0.368Pkt Len Var + 0.346Init Fwd Win Byts - 0.335Init Bwd Win Byts + …
23	0.66 Fwd Seg Size Min - 0.638Fwd Header Len - 0.157Idle Std + …

These acquired features have been rated based on the feature effectiveness. Therefore, these features that have been extracted based on PCA are sorted so that the feature, with more impact, has more value. This ranking was performed by the Filter method [29] Based on their evaluations. This ranking value is shown in Fig. 7.

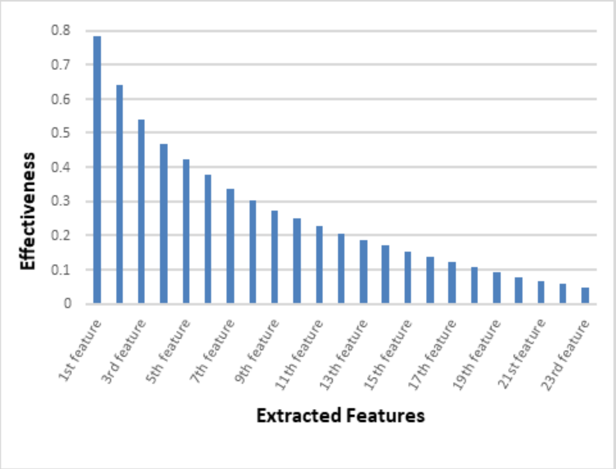


Fig. 7 Sorted extracted PCA features

The processed dataset is ready for training and evaluation in an attack detection system.

4-3- System Setup in SDN

Table 2 shows the specifications of the implementation environment. The Mininet emulator environment, which is licensed under the open-source BSD, is used to implement the Software Defined Network. Mininet provides a virtual environment in which all programs, switches, and code running on the actual system kernel, which can be a virtual machine, a cloud system, or a local system. The data layer uses the Open vSwitch (OVS). OVS is a multi-layer virtual switch that is Apache certified. These switches are programmable and support the OpenFlow protocol

Table 2: Environment Information

Name	Type / Name
OS	Ubuntu 18.04 64bit
CPU	Intel® Core™ i7-7700HQ CPU @ 2.80GHz × 3
RAM	8 GB
Simulator	Mininet
Switch	openvswitch 2.9.2
Controller	Floodlight Master
South API	OpenFlow1.3

In the control layer, the Floodlight Master controller is used in the network control section. Floodlight [30] is a Java-based open-source controller that supports both OpenFlow physical and virtual switches. This controller is Apache certified and has good scalability. The Floodlight Master version supports the architecture of distributed controllers efficiently. Fig. 8 shows the topology used in this scenario. One of these hosts is a Simple HTTP Server for web service that is considered as a victim of a DDoS attack. Of the other three hosts, one is considered as an attacker and the other as a normal user.

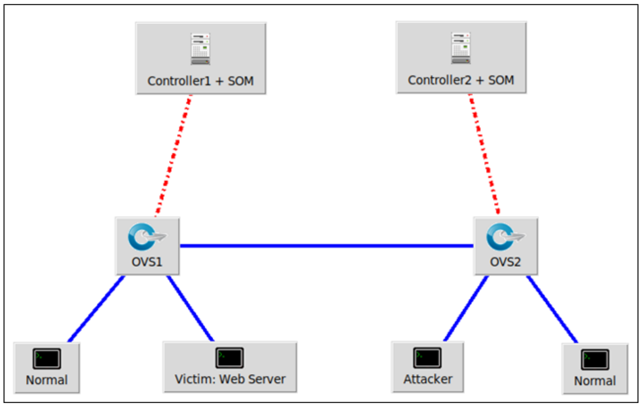


Fig. 8 Scenario Topology

Syn Flooding attack is used for the test. Scapy was used on the attacker host to implement this attack. To generate normal traffic, a shell script is used, which is sent to the webserver at the same time as the attack traffic.

In this architecture, the whole network can be divided into several independent domains in terms of geography and management. In our topology, the whole network is divided into two domains and each domain is managed by one controller. This architecture consists of two main parts:

1) Intra-domain communication, which includes the main function of the controller, i.e. sending policies and rules to the switches and receiving their status through the southbound interface.
2) Inter-domain communication, which includes the synchronization between the controllers through the east-west API as shown in Fig. 9.

Controller1:

```
[o.s.s.i.SyncManager] [1] Updating sync configuration ClusterConfig [allNodes={1=Node [hostname=127.0.0.1, port=6643, nodeId= domainId=1], 2=Node [hostname=127.0.0.1, port=6644, nodeId=2, domainId=2]}, authScheme=NO_AUTH]
[o.s.s.i.SyncManager] [1->2] Synchronizing local state to remote node
```

Controller2:

```
[o.s.s.i.SyncManager] [2] Updating sync configuration ClusterConfig [allNodes={1=Node [hostname=127.0.0.1, port=6643, nodeId=1, domainId=1], 2=Node [hostname=127.0.0.1, port=6644, nodeId=2, domainId=2]}, authScheme=NO_AUTH]
[o.s.s.i.SyncManager] [2->1] Synchronizing local state to remote node]
```

Fig. 9 Controller Synchronization

4-4- Flow Collection in Experiment

In the proposed method, the attack detection takes place in the control layer; That is, from the messages sent between the switch and the controller, abnormal flows related to the distributed denial-of-service attack must be identified. The challenge is that not all traffic packets pass through the control layer of the SDN network. To achieve this information, the flow entry in the flow table of each switch is used.

To solve the above challenge, the statistical information of network flows must be extracted from the messages exchanged in the OpenFlow protocol. The Floodlight controller consists of several components. One of these components is related to the collection of information from flow tables of network switches. This component sends a request to the switches at predetermined intervals, and the switches respond to the flow table information in response. Determining the amount of time interval, it takes to send a request is very important. If this time interval is considered too long, there will be a long delay in detecting the attack and if considered too short, the number of requests and responses between the controller and the switches will increase, increasing overhead. According to the paper [31], the time interval of the request is considered 3 seconds. Now, using this information, each controller will have its domain flow statistic.

4-5- Analysis of Detection Method

The self-organizing map consists of two layers, the input layer and the neural network layer. The number of neurons in the neural network layer indicates the number of output clusters that cluster the training data due to the unsupervised nature of this algorithm. This mode is useful when the training data is unlabeled and it is not clear to which category each input belongs. The data used in this article are labelled and fall into two general classes: Benign and DDoS. Therefore, the output of the detection system must be such that it can eventually map network neurons into these two classes. For this purpose, in addition to a self-organizing map, a supervised learning method called Learning Vector Quantization (LVQ) has been used. LVQ networks are a special type of competitive neural network that uses the idea of supervised learning and their main application is in classifying and recognizing patterns. This network is the development of a self-organizing map in a supervised state, and its learning method is quite similar to a self-organizing map, except that in LVQ only the winning neurons are moved and tuned each time, while in the self-organizing map, in addition to the winning neurons, the neighbouring neurons also move slightly. When this method is used in conjunction with a self-organizing map, it does the training twice, first clustering it unsupervised by the self-organizing map and then classifying it by learning vector Quantization. Fig. 10 shows the LVQ-SOM process. In this figure, from left to right, the first two layers are related to SOM mapping and the last two layers are related to LVQ.

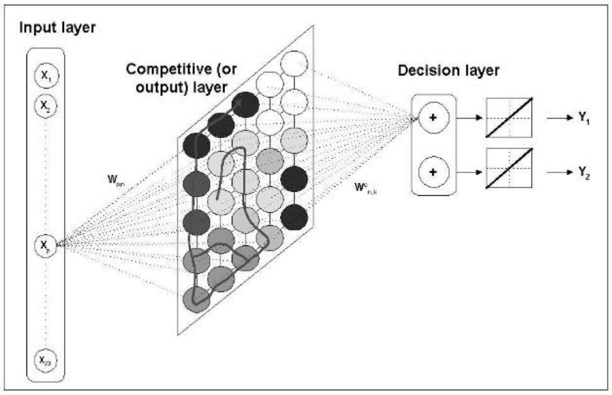


Fig. 10 LVQ-SOM [14]

The characteristics used in LVQ-SOM Learning are listed in Table 3.

Table 3: LVQ-SOM Parameters

Parameters	Value
Learning Rate	0.1
Epoch Limit	1440
Distance Type	Euclidean

5- Results and Analysis

To train the proposed artificial neural network is done with a dataset including different types of DDoS attacks as if it can discriminate varying DDoS types. The DDoS attacks which have been considered in this paper are SYN, DNS, LDAP, NTP and UDP attacks. Table 4 shows DDoS attack types distribution in this dataset. The number of attacked traffic and normal traffics are balanced in the training set. In this research, the DDoS attacks are not discriminated; hence, all DDoS attacks are combined as DDoS. To improve the training phase, the training set has been shuffled.

Table 4: DDoS Attacks for Training and Testing

Attack Type	# of Flows
SYN	30321
DNS	1985
LDAP	4007
NTP	13479
UDP	1119

In machine learning methods, evaluation metrics are divided into two stages; the Training phase and the test phase. In the training phase, evaluation metrics are used to optimize the algorithm. In other words, evaluation metrics are used to select the best solution to increase the estimation accuracy of the algorithm. While in the test phase, the evaluation metrics measure the efficiency of the model created in the classification of new data.

5-1- Evaluation Metrics

One of the most important evaluation metrics in two-tier classes is the Confusion Matrix. This matrix is 2 × 2, the rows of which represent the estimated categories and the columns of which represent the actual classes. In this matrix, four variables are defined, which are summarized in Table 5 as shown below.

Table 5: Evaluation variables

Evaluation variables	Definition
True Positive (TP)	Attack traffic identified as an attack
False Positive (FP)	Attack traffic detected as normal
True Negative (TN)	Normal traffic that is identified as normal
False Negative (FN)	Normal traffic identified as an attack

These variables can be used to derive other evaluation metrics:

- **Accuracy** shows the ratio of correct estimates to the total.

$$Acc = \frac{tp+tn}{tp+fp+tn+fn} \qquad (10)$$

- **Error Rate** that shows the ratio of incorrect estimates to the total.

$$Err = \frac{fp+fn}{tp+fp+tn+fn} \qquad (11)$$

- **Precision** indicates the ratio of positively estimated in positive class to total positive class. This criterion is used when the false positive rate is significant.

$$P = \frac{tp}{tp+fp} \qquad (12)$$

- **Recall** is the ratio of positively estimated in positive class to the total number of samples estimated as positive. This criterion is used when the false negative is significant.

$$R = \frac{tp}{tp+fn} \qquad (13)$$

- **F-Measure** is the harmonic mean of precision and recall.

$$FM = \frac{2 \times P \times R}{P+R} \qquad (13)$$

- **Mean Square Error (MSE)** is the metric for evaluating the error of the training step

$$MSE = \frac{1}{n}\sum_{j=1}^{n}(P_j - A_j)^2 \qquad (14)$$

The evaluation metrics are summarized in Table. 6. These metrics are used to compare our proposed model with the other similar approach.

Table 6: Evaluation metrics

Evaluation Metrics	Definition
Accuracy	The ratio of correct estimates to the total
Error rate (ER)	The ratio of incorrect estimates to the total
Precision	The ratio of positively estimated in positive class to the total positive class
Recall	The ratio of positively estimated in positive class to the total number of samples estimated as positive
F-Measure	Harmonic Mean of precision and recall

5-2- Performance Evaluation

One of the most important characteristics of neural networks is the number of neurons. Increasing the number of neurons does not necessarily increase the accuracy of the diagnosis. If the number of neurons is more than a certain limit, they will not improve the accuracy and even reduce the accuracy. Therefore, the number of neurons must be selected in such a way as to maximize the accuracy of the neural network. There is no specific method or formula for determining the number of neurons and it can only be obtained experimentally.

The efficiency of the neural network used in the proposed method has been evaluated with several different neurons. It started with 100 neurons and continued until 2000 neurons. If the accuracy and error rate metrics are important, according to the diagrams in Fig. 11 and Fig. 12 of the case where the number of neurons is 1500, the accuracy of the neural network is 98.86%, which is the highest compared to other cases. Also, with this number of neurons, the error rate becomes 1.01%, which is the lowest error rate. We hope that our research can address some of the security challenges in SDN.

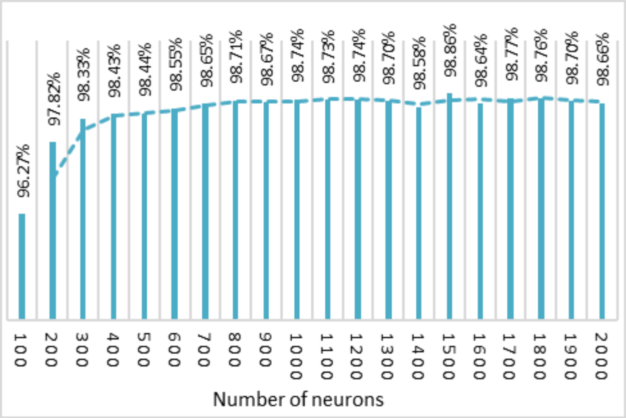


Fig. 11 Accuracy Rate for Different Number of Neurons

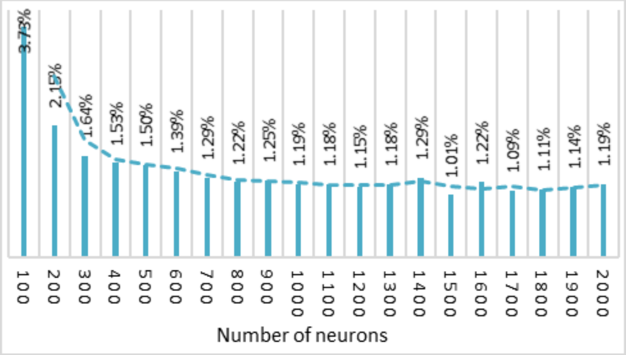


Fig. 12. Error Rate for Different Number of Neurons

If the metrics for precision and recall are important, according to the diagram in Fig. 13, which is drawn based on the F-measure, which is the equivalent of the two metrics for precision and recall, while the number of neurons is considered to be 1200 will have the most value. If the value of the precision metric is high, it means that the flows which are low likely to be attacked are recognized as normal flows. On the other hand, if the value of the recall metric is high, it means that a huge number of the normal flows are recognized as attack flows.

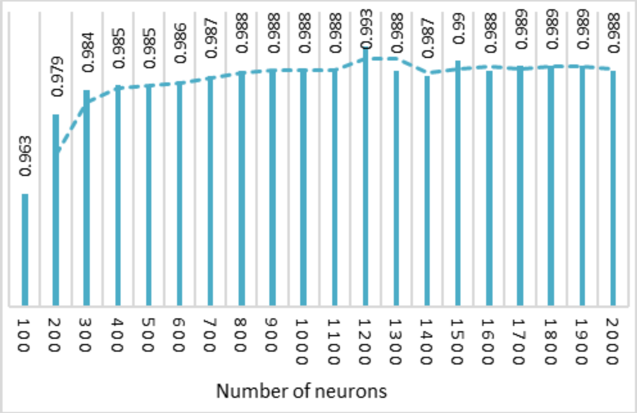


Fig. 13 F-Measure for Different Number of Neurons

Table 7 shows the evaluation metrics calculated in the proposed method for the number of different neurons.

Table 7: CICDDoS2019 Datasets Features

	Acc	Err	Precision	Recall	F-Measure	MSE
100	96.27%	3.73%	0.958	0.968	0.963	0.1683
200	97.82%	2.15%	0.973	0.984	0.979	0.1315
300	98.33%	1.64%	0.978	0.99	0.984	0.1134
400	98.43%	1.53%	0.98	0.989	0.985	0.1077
500	98.44%	1.5%	0.98	0.99	0.985	0.107
600	98.55%	1.39%	0.981	0.992	0.986	0.1025
700	98.65%	1.29%	0.984	0.99	0.987	0.099
800	98.71%	1.22%	0.985	0.991	0.988	0.0955
900	98.67%	1.25%	0.982	0.993	0.988	0.097
1000	98.37%	1.19%	0.985	0.997	0.988	0.0952
1100	98.73%	1.18%	0.986	0.991	0.988	0.0938
1200	98.74%	1.15%	0.984	0.993	0.993	0.0935
1300	98.7%	1.18%	0.985	0.991	0.988	0.0944
1400	95.58%	1.29%	0.983	0.991	0.987	0.0972
1500	98.86%	1.01%	0.988	0.991	0.99	0.0884
1600	98.64%	1.22%	0.984	0.992	0.988	0.0944
1700	98.77%	1.09%	0.985	0.993	0.989	0.0912
1800	98.76%	1.11%	0.986	0.992	0.989	0.0909
1900	98.7%	1.14%	0.984	0.993	0.989	0.0922
2000	98.66%	1.19%	0.983	0.993	0.988	0.0933

In order to better display the output of the work, we have made a comparison with one of the similar works. Nam et al. in [28] proposed two classification mechanisms to detect DDoS attacks in SDN with centralized controller architecture. These mechanisms are SOM + KNN and SOM with center-distributed classification and the features are Entropy of source IP, Entropy of source port, Entropy of destination port, Entropy of packet, protocol and the

Total number of packets. In this work, the monitor module collects the traffic information from the switches and after processing forward to the Algorithm module. The algorithm module classifies the network state as normal or under attack. If the network is under attack, then it generates an alert to the mitigation module. Then, the mitigation module generates the new policies and forwards these decisions to the switches as well as the server.

Our comparison was evaluated through the Detection Rate measurement (DR) and the False Alarm rate (FA), computed using Equations 15 and 16, respectively.

$$DR = \frac{tp}{tp + fn} \qquad (15)$$

$$FA = \frac{fp}{tn + fp} \qquad (16)$$

Table. 8 shows the results of our proposed model and SOM + KNN and SOM with center-distributed classification.

Table 8: Comparison results

Method	DR(%)	FA(%)
Proposed model	99.56	0.86
SOM + KNN	98.24	2.14
SOM with center-distributed	97.28	22.36

6- Conclusion

The main idea in this paper is a novel model to detect the DDoS attack, so a Self-Organizing Map (SOM) has been used to cluster the traffic datasets according to their similarity. The model has been designed and evaluated with the CICDDoS2019 dataset that had been labelled before. The simulation has been implemented in WEKA and the results show that the SOM works well with 1500 neurons. According to the labelled CICDDoS2019, the similarity is about 98.3% which is acceptable. To define DDoS traffic detection, a novel model has been proposed. The feature extraction has been done with the PCA method and trained with the CICDDoS2019 and LVQ. The proposed DDoS attack detection model has been developed in the controller that is the main layer of SDN architecture. The other momentous contribution is that the proposed model could protect the network from DDoS attacks with the distributed controllers' structure. The DDoS attack detection model has been implemented in the Floodlight controller in java language using the WEKA library. The simulation has been done with Mininet as an SDN emulation. The simulation results indicate that the proposed model could reach 99.56% accuracy to detect DDoS attacks while this model has been implemented in SDN architecture with distributed controllers. The proposed model has reached an acceptable accuracy, but the drawback of this model is the time consumption for clustering the traffics and limited dataset which can be the future work. The computation time can be considered as another future work.

References

[1] A. Shirmarz and A. Ghaffari, "An Autonomic Software Defined Network (SDN) Architecture With Performance Improvement Considering," J. Inf. Syst. Telecommun., vol. 8, no. 2, pp. 1–9, 2020.

[2] A. Shirmarz and A. Ghaffari, "Performance issues and solutions in SDN-based data center: a survey," J. Supercomput., 2020.

[3] A. Shirmarz and A. Ghaffari, "An adaptive greedy flow routing algorithm for performance improvement in a software- defined network," Int. Numer. Model. Electron. networks, Devices, Fields-Wiley online Libr., no. March, pp. 1–21, 2019.

[4] R. Masoudi and A. Ghaffari, "Software defined networks: A survey," J. Netw. Comput. Appl., vol. 67, pp. 1–25, 2016.

[5] Z. Zhao et al., "Autonomic communications in software-driven networks," IEEE J. Sel. Areas Commun., vol. 35, no. 11, pp. 2431–2445, 2017.

[6] A. Shirmarz and A. Ghaffari, "Taxonomy of controller placement problem (CPP) optimization in Software Defined Network (SDN): a survey," J. Ambient Intell. Humaniz. Comput., no. 0123456789, 2021.

[7] A. G. Alireza Shirmarz, "Automatic Software Defined Network (SDN) Performance Management Using TOPSIS Decision-Making Algorithm," J. Grid Comput., 2021.

[8] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," Proc. - Int. Carnahan Conf. Secur. Technol., vol. 2019-Octob, 2019.

[9] T. Cisco and A. Internet, "Cisco Annual Internet Report," 2020.

[10] "Legal Implications of DDoS Attacks and the Internet of Things (IoT)," 2016. [Online]. Available: https://www.dataprotectionreport.com/2016/12/legal-implications-of-ddos-attacks-and-the-internet-of-things-iot/.

[11] "Defending against Distributed Denial of Service (DDoS) attacks," 2020. [Online]. Available: https://www2.deloitte.com/ca/en/pages/risk/articles/DDoSattacks.html.

[12] "UNB Dataset." [Online]. Available: www . unb . ca / cic / datasets / ddos - 2019 . html.

[13] Q. Niyaz, W. Sun, and M. Alam, "Impact on SDN Powered Network Services Under Adversarial Attacks," Procedia - Procedia Comput. Sci., vol. 62, no. Scse, pp. 228–235, 2015.

[14] Teuvo Kohonen, The Basic SOM. 2001.

[15] T. V Phan, N. K. Bao, and M. Park, "Author ' s Accepted Manuscript Performance Bottleneck Handler for Large-sized Software- Defined Networks under Flooding Attacks Reference : Distributed-SOM : A Novel Performance Bottleneck Handler for Large-sized," J. Netw. Comput. Appl., 2017.

[16] Teuvo Kohonen, "The self-organizing map," in Proceedings of the IEEE, 1990, pp. 1464–1480.

[17] B. Yuan, D. Zou, S. Yu, H. Jin, W. Qiang, and J. Shen, "Defending against flow table overloading attack in software-defined networks," IEEE Trans. Serv. Comput., vol. 12, no. 2, pp. 231–246, 2019.

[18] M. Clayton, C. Batt, M. Clayton, and C. Batt, Communications and networking. 2019.

[19] M. Xuanyuan, V. Ramsurrun, and A. Seeam, “Detection and mitigation of DDoS attacks using conditional entropy in software-defined networking,” Proc. 11th Int. Conf. Adv. Comput. ICoAC 2019, pp. 66–71, 2019.

[20] A. Ahalawat, S. S. Dash, A. Panda, and K. S. Babu, “Entropy Based DDoS Detection and Mitigation in OpenFlow Enabled SDN,” Proc. - Int. Conf. Vis. Towar. Emerg. Trends Commun. Networking, ViTECoN 2019, pp. 1–5, 2019.

[21] S. M. Mousavi and M. St-hilaire, “Early Detection of DDoS Attacks against SDN Controllers,” in International Conference on Computing, Networking and Communications, Communications and Information Security Symposiu, 2015, pp. 77–81.

[22] S. M. S. Mousavi and M. St-Hilaire, “Early Detection of DDoS Attacks in Software Defined Networks Controller,” Thesis, pp. 77–81, 2014.

[23] M. Wang, Y. Lu, and J. Qin, “A dynamic MLP-based DDoS attack detection method using feature selection and feedback,” Comput. Secur., vol. 88, p. 101645, 2020.

[24] T. V. Phan, N. K. Bao, and M. Park, “Distributed-SOM: A novel performance bottleneck handler for large-sized software-defined networks under flooding attacks,” J. Netw. Comput. Appl., vol. 91, pp. 14–25, 2017.

[25] R. Braga, E. Mota, and A. Passito, “Lightweight DDoS flooding attack detection using NOX/OpenFlow,” in Proceedings - Conference on Local Computer Networks, LCN, 2010, pp. 408–415.

[26] A. Detection, S. Networking, and S. K. Dey, “Effects of Machine Learning Approach in Flow-Based,” 2019.

[27] R. Santos, D. Souza, W. Santo, A. Ribeiro, and E. Moreno, “Machine learning algorithms to detect DDoS attacks in SDN,” Concurr. Comput. , vol. 32, no. 16, pp. 1–14, 2020.

[28] T. M. Nam et al., “Self-organizing map-based approaches in DDoS flooding detection using SDN,” Int. Conf. Inf. Netw., vol. 2018-Janua, pp. 249–254, 2018.

[29] T. Khalil, “A Survey of Feature Selection and Feature Extraction Techniques in Machine Learning,” pp. 372–378, 2014.

[30] S. Rowshanrad, V. Abdi, and M. Keshtgari, “Performance evaluation of SDN controllers: Floodlight and Opendaylight,” Int. Islam. Univ. Malaysia Eng. J., vol. 17, no. 2, pp. 47–57, 2016.

[31] R. Braga, E. Mota, and A. Passito, “Lightweight DDoS Flooding Attack Detection Using NOX/ OpenFlow,” in 35th Annual IEEE Conference on Local Computer Networks, 2010, no. January 2015

Journal of Information Systems and Telecommunication
Vol.10, No.2, April-June 2022, 132-140

http://jist.acecr.org
ISSN 2322-1437 / ESSN:2345-2773

Defect Detection using Depth Resolvable Statistical Post Processing in Non-Stationary Thermal Wave Imaging

G.V.P. Chandra Sekhar Yadav[1,2*], V.S. Ghali[1], Naik. R. Baloji[3]

[1].Infrared Imaging Center, Department of ECE, Koneru Lakshmaiah Education Foundation, A.P, Vaddeswaram, India
[2].Department of ECE, DVR & Dr HS MIC College of Technology, Kanchikacherla, A.P, India
[3].Naval Materials Research Laboratory, Ambernath (E), Dist. Thane, Maharashtra, 421506 India

Received: 31 Jan 2021/ Revised: 25 Oct 2021/ Accepted: 05 Dec 2021
DOI:

Abstract

Defects that are generated during various phases of manufacturing or transporting limit the future applicability and serviceability of materials. In order to detect these defects a non-destructive testing modality is required. Depth resolvable subsurface anomaly detection in non-stationary thermal wave imaging is a vital outcome for a reliable prominent investigation of materials due to its fast, remote and non-destructive features. The present work solves the 3-Dimensional heat diffusion equation under the stipulated boundary conditions using green's function based analytical approach for recently introduced quadratic frequency modulated thermal wave imaging (with FLIR SC 655A as infrared sensor with spectral range of 7.5-14μm and 25 fps) to explore the subsurface details with improved sensitivity and resolution. The temperature response obtained by solving the 3-Dimensional heat diffusion equation is used along with random projection-based statistical post-processing approach to resolve the subsurface details by imposing a band of low frequencies (0.01-0.1 Hz) over a carbon fiber reinforced polymer for experimentation and extracting orthonormal projection coefficients to improve the defect detection with enhanced depth resolution. Orthonormal projection coefficients are obtained by projecting the orthonormal features of the random vectors that are extracted by using Gram-Schmidt algorithm, on the mean removed dynamic thermal data. Further, defect detectability of random projection-based post-processing approach is validated by comparing the full width at half maxima (FWHM) and signal to noise ratio (SNR) of the processed results of the conventional approaches. Random projection provides detailed visualization of defects with 31% detectability even for deeper and small defects in contrast to conventional post processing modalities. Additionally, the subsurface anomalies are compared with their sizes based on full width at half maxima (FWHM) with a maximum error of 0.99% for random projection approach.

Keywords: Non-Stationary Thermal Wave Imaging (NSTWI); Fast Fourier Transform (FFT); Correlation; Random Projection Transform (RPT).

1- Introduction

Subsurface analysis for non-stationary thermal wave imaging (NSTWI) gaining importance from the past decades due to its distinct wide, fast, non-invasive and remote testing properties. Different processing techniques are available to get the hidden features of the material. Out of them, the conventional phase approach is mostly used due to reduced non-uniform emissivity and radiation. Even though the conventional phase-based analysis gives the subsurface details, but it is not used to get deeper details due to its limited frequency resolution. This paper focuses on the recently introduced depth resolvable post-processing approach to provide fine subsurface details with improved frequency resolvable NSTWI technique.

Active infrared thermography is one of the non-destructive testing (NDT) methods to test the integrity of the material,[1] without impairing its future utility. In active infrared thermography, subsurface analysis is done by comparing the thermal response obtained by variations in thermophysical properties of diffused waves at anomalies of different depths. Pulse thermography (PT), lock-in thermography (LT), pulse phase thermography (PPT) and non-stationary thermography are various techniques of active thermography classified on the employment of optical input.

A large power rectangular input with less duration is used to test the integrity of the object in PT,[2] but the obligation of more power along with non-uniform effects limits it. In

🖂 G.V.P. Chandra Sekhar Yadav
sekhar.yadav2008@gmail.com

LT, a mono frequency continuous sinusoidal stimulus with small peak is used for investigation and phase-based analysis is performed to get the subsurface details. Limited depth resolution with mono frequency input stimulus,[3] causes repetitive experimentation in LT. In PPT, a similar stimulus is applied as in PT and phase analysis is done to get the subsurface details like LT to reduce the non-uniform effects; yet the obligation of more power [4] remained as a constraint of conventional thermographic approaches. Thus, NSTWI is introduced to test the integrity of the object with low power non-stationary stimulus in single experimentation.[5, 6, 7, 8, 9, 10, 11, 12]

In 2006, R. Mulaveesala introduced linear frequency modulated thermal wave imaging (LFMTWI),[5] which uses a low power continuous band of low frequency modulated chirped stimulus to test the object with moderate depth resolution. Later V.S. Ghali introduced its quadratic version called quadratic modulated thermal wave imaging (QFMTWI) in 2012,[6] by imposing a continuous band of low quadratic frequency-modulated chirped stimulus over the object with improved depth resolvable capability.

In the conventional phase approach, a fast fourier transform (FFT) with the orthogonal basis of multiple frequencies is applied on every thermal profile, later their relative phase value is extracted to discriminate,[7] the defective and non-defective regions. Pulse compression (PC) uses a matched filter-based cross-correlation approach and corresponding peak delays,[8, 9] are used to characterize defective and non-defective regions. In the Random projection transform (RPT) approach, high-dimensional data can be subsampled using the Johnson-Lindenstrauss lemma, while maintaining its maximum features by projecting,[10] on a smaller number of orthonormal random basis vectors with less complexity.

This work uses QFMTWI for experimental investigation of carbon fiber reinforced polymer (CFRP). Later, RPT based post-processing technique uses the coefficients of orthonormal projection to discriminate the defective and non-defective regions of the test object. This paper validates the defect detection capability of RPT technique by considering full width at half maxima (FWHM) and signal to noise ratio (SNR) and comparing them with conventional FFT and PC.

2- Theory of Thermal Waves

The temperature response on the surface of the material has been obtained by solving the 3-D heat equation by considering internal heat generation Q_m. The mathematical model is obtained by using Greens function under the Neumann boundary conditions.[11] Consider 3-D diffusion Eq. (1)

$$\frac{\partial^2 T(x,y,z,t)}{\partial^2 x}+\frac{\partial^2 T(x,y,z,t)}{\partial^2 y}+\frac{\partial^2 T(x,y,z,t)}{\partial^2 z}=\frac{1}{\alpha}\frac{\partial T(x,y,z,t)}{\partial t}+Q_m \tag{1}$$

Where α refers the thermal diffusivity of the test specimen. The solution is acquired by resolving the homogeneous section of Eq. (2) using variable separable method and it can be written as

$$T(x,y,z,t)=X(x)Y(y)Z(z)\Gamma(t) \tag{2}$$

By substituting Eq. (2) in Eq. (1)

$$\frac{\partial^2 (X(x)Y(y)Z(z)\Gamma(t))}{\partial^2 x}+\frac{\partial^2 (X(x)Y(y)Z(z)\Gamma(t))}{\partial^2 y}$$

$$+\frac{\partial^2 (X(x)Y(y)Z(z)\Gamma(t))}{\partial^2 z}=\frac{1}{\alpha}\frac{\partial X(x)Y(y)Z(z)\Gamma(t))}{\partial t} \tag{3}$$

$$\frac{Y(y)Z(z)\Gamma(t)\partial^2 X(x)}{\partial^2 x}+\frac{X(x)Z(z)\Gamma(t)\partial^2 Y(y)}{\partial^2 y}$$

$$+\frac{X(x)Y(y)\Gamma(t)\partial^2 Z(z)}{\partial^2 z}=\frac{1}{\alpha}\frac{X(x)Y(y)Z(z)\partial\Gamma(t)}{\partial t} \tag{4}$$

Divide with $X(x)Y(y)Z(z)\Gamma(t)$ on both sides

$$\frac{X''(x)}{X(x)}+\frac{Y''(y)}{Y(y)}+\frac{Z''(z)}{Z(z)}=\frac{1}{\alpha}\frac{\Gamma'(t)}{\Gamma(t)} \tag{5}$$

Let

$$\frac{X''(x)}{X(x)}=-\beta^2;\frac{Y''(y)}{Y(y)}=-\gamma^2;\frac{Z''(z)}{Z(z)}=-\eta^2;\frac{\Gamma'(t)}{\Gamma(t)}=-\lambda^2 \tag{6}$$

Put Eq. (6) in Eq. (5), we get

$$\lambda^2=\alpha(\beta^2+\gamma^2+\eta^2) \tag{7}$$

Since the solutions are negative

$$X(x)=A_1\cos\beta x+B_1\sin\beta x$$

$$Y(y)=A_2\cos\gamma y+B_2\sin\gamma y$$

$$Z(z)=A_3\cos\eta z+B_3\sin\eta z$$

$$\Gamma(t)=Ce^{-\alpha(\beta^2+\gamma^2+\eta^2)t} \tag{8}$$

Substitute Eq. (8) in Eq. (2)

$$T(x,y,z,t)=(A_1\cos\beta x+B_1\sin\beta x)$$

$$(A_2\cos\gamma y+B_2\sin\gamma y)(A_3\cos\eta z+B_3\sin\eta z)$$

$$Ce^{-\alpha(\beta^2+\gamma^2+\eta^2)t} \tag{9}$$

Boundary conditions and initial conditions are

$$-k\frac{\partial T}{\partial x}\Big|_{x=0}=Q_T(t);-k\frac{\partial T}{\partial x}\Big|_{x=a}=0;$$

$$-k\frac{\partial T}{\partial y}\Big|_{y=0,b}=0;$$

$$-k\frac{\partial T}{\partial z}\Big|_{z=0,c}=0;$$

$$T(x,y,z,0)=T_0; \tag{10}$$

Where $Q_T(t)$ is the combination of DC and QFM chirp signal i.e., $Q_T(t)=1+Q(t)$

Q(t) is the QFM chirp signal; 1 is the DC added to QFM to overcome the simultaneous heating and cooling phase of experimentation $Q_T(t) = Q_0\left(1+e^{j2\pi(f_0+b_0t^2)t}\right); x = 0$

Where, Q_0 refers constant heat flow, f_0 refers starting frequency and b_0 refers the chirp rate of the incident QFM chirp signal. Fig. 1 represents the DC added QFM chirp signal of 2kw power at 0.01-0.1Hz frequency.

By substituting Eq. (10) in Eq. (9), we get solution for Greens function for Eq. (1) [11]

$$G(x,y,z,t;\varepsilon,\xi,\delta,\tau) =$$

$$\frac{R_1R_2R_3}{abc}\sum_{m=0}^{\infty}\sum_{n=0}^{\infty}\sum_{p=0}^{\infty}\cos\beta_m x\cos\beta_m\varepsilon$$

$$\cos\gamma_n y\cos\gamma_n\xi\cos\eta_p z\cos\eta_p\delta$$

$$e^{-\alpha(\beta_m^2+\gamma_n^2+\eta_p^2)(t-\tau)} \qquad (11)$$

$$\beta_m = \frac{m\pi}{a}; m = 0,1,2....\infty$$

$$\gamma_n = \frac{n\pi}{b}; n = 0,1,2....\infty$$

Where $\eta_p = \frac{p\pi}{c}; p = 0,1,2....\infty$

$$R_1 = \begin{cases}1; m=0\\ 2; m=1,2.......\infty\end{cases}$$

$$R_2 = \begin{cases}1; n=0\\ 2; n=1,2.......\infty\end{cases}$$

$$R_3 = \begin{cases}1; p=0\\ 2; p=1,2.......\infty\end{cases}$$

In terms of Greens function the solution for Eq. (1) which refers non-homogeneous heat diffusion is given as

$$T(x,y,z,t) = \begin{array}{l}\frac{\alpha}{k}\int_0^t\int_0^a\int_0^b\int_0^c Q_m G(x,y,z,t;\varepsilon,\xi,\delta,\tau)d\delta d\xi d\varepsilon d\tau\\ -\alpha\int_0^t\int_0^b\int_0^c T(\varepsilon,\xi,\delta,\tau)\nabla_j G(x,y,z,t;\varepsilon,\xi,\delta,\tau)d\delta d\xi\\ +\alpha\int_0^t\int_0^b\int_0^c G(x,y,z,t;\varepsilon,\xi,\delta,\tau)\nabla_j T(\varepsilon,\xi,\delta,\tau)d\delta d\xi\\ +\int_0^a\int_0^b\int_0^c T(\varepsilon,\xi,\delta,0)G(x,y,z,t;\varepsilon,\xi,\delta,0)d\delta d\xi d\varepsilon \qquad (12)\end{array}$$

$$T(x,y,z,t) = \begin{array}{l}\frac{\alpha}{k}\int_0^t\int_0^a\int_0^b\int_0^c Q_m G(x,y,z,t;\varepsilon,\xi,\delta,\tau)d\delta d\xi d\varepsilon d\tau\\ +\alpha\int_0^t\int_0^b\int_0^c G(x,y,z,t;0,\xi,\delta,\tau)\frac{\partial T(0,\xi,\delta,\tau)}{\partial x}d\delta d\xi d\tau\\ -\alpha\int_0^t\int_0^b\int_0^c G(x,y,z,t;a,\xi,\delta,\tau)\frac{\partial T(a,\xi,\delta,\tau)}{\partial x}d\delta d\xi d\\ +\int_0^a\int_0^b\int_0^c T(\varepsilon,\xi,\delta,0)G(x,y,z,t;\varepsilon,\xi,\delta,0)d\delta d\xi d\varepsilon \qquad (13)\end{array}$$

Apply the boundary conditions and initial conditions from Eq. (10) to above Eq. (13), we get

$$T(x,y,z,t) = \begin{array}{l}\frac{\alpha}{k}\int_0^t\int_0^a\int_0^b\int_0^c Q_m G(x,y,z,t;\varepsilon,\xi,\delta,\tau)d\delta d\xi d\varepsilon d\tau\\ -\frac{\alpha}{k}\int_0^t\int_0^b\int_0^c G(x,y,z,t;0,\xi,\delta,\tau)Q_T(\tau)d\delta d\xi d\tau\\ +\int_0^a\int_0^b\int_0^c T(\varepsilon,\xi,\delta,0)G(x,y,z,t;\varepsilon,\xi,\delta,0)d\delta d\xi d \qquad (14)\end{array}$$

In the case of experimentation $Q_T(t)$ is considered as the combination of DC and QFM chirp signal to overcome the simultaneous heating and cooling process.

After the experimentation while doing the qualitative analysis, DC response was removed as it doesn't contain any information.

Hence the final temperature equation can be obtained from Eq. (14) is

$$T(x,y,z,t) = -\frac{\alpha}{k}\int_0^t\int_0^b\int_0^c G(x,y,z,t;0,\xi,\delta,\tau)Q(\tau)d\delta d\xi d\tau \qquad (15)$$

By substituting Greens function from Eq. (11) the above Eq. (15) can be written as

$$T(x\ ,y,z,t) = -\frac{\alpha}{k}\int_0^t\int_0^b\int_0^c\frac{R_1R_2R_3}{abc}$$

$$\sum_{m=0}^{\infty}\sum_{n=0}^{\infty}\sum_{p=0}^{\infty}\cos\beta_m x\cos\gamma_n y\cos\gamma_n\xi$$

$$\cos\eta_p z\cos\eta_p\delta e^{-\alpha(\beta_m^2+\gamma_n^2+\eta_p^2)(t-\tau)}Q(\tau)d\delta d\xi d\tau \qquad (16)$$

$$T(x,y,z,t) = -\frac{\alpha}{k}\int_0^t\frac{R_1}{a}\sum_{m=0}^{\infty}\cos\beta_m x e^{-\alpha(\beta_m^2)(t-\tau)}Q(\tau)d\tau \qquad (17)$$

$$T(x,y,z,t) = -\frac{\alpha}{ka}\int_0^t\left(1+2\sum_{m=1}^{\infty}\cos\beta_m x e^{-\alpha(\beta_m^2)(t-\tau)}\right)Q(\tau)d\tau \qquad (18)$$

By considering m=0 and 1 and neglecting higher order m values because

$\cos\beta_m x = 1; m = 0,1,2,3....\infty$ for $a = 0.005m$

$$e^{-\alpha(\beta_m^2)(t-\tau)} = e^{-0.42*10^{-6}\left(\frac{m\pi^2}{25*10^{-6}}\right)(t-\tau)} \begin{array}{ll} =1 & ;m=0 \\ =0.84 & ;m=1 \\ =0.515 & ;m=2 \\ =0.22 & ;m=3 \\ =0.07 & ;m=4 \\ =0.015 & ;m=5 \\ =0.002 & ;m=6 \\ =0.0002 & ;m=7 \\ =0.00002 & ;m=8 \end{array}$$

On the application of Laplace transform, we get

$$T(X,Y,Z,s) = -\frac{\alpha}{ka}\frac{Q(s)}{s} + \frac{2\alpha\cos\beta_m x}{ka}\int_0^t\left(\sum_{m=1}^{\infty} e^{-\alpha(\beta_m^2)(t-\tau)}\right)Q(\tau)d\tau$$

$$T(X,Y,Z,s) = -\frac{\alpha}{ka}\frac{Q(s)}{s} + \frac{2\alpha\cos\left(\frac{\pi}{a}x\right)}{ka}\int_0^t e^{-\frac{\alpha(\pi^2)(t-\tau)}{a^2}} Q(\tau)d\tau$$

$$T(X,Y,Z,s) = -\frac{\alpha}{ka}\frac{Q(s)}{s} + \frac{2\alpha\cos\left(\frac{\pi}{a}x\right)}{ka}\frac{e^{-\frac{\alpha\pi^2 t}{a^2}}Q\left(s-\frac{\alpha\pi^2}{a^2}\right)}{s} \qquad (19)$$

Where Q(s) refers the Laplace transform of Q(t) at x=0, α is thermal diffusivity of the test sample, k refers the thermal conductivity of the material and a is the length in x-direction respectively. Further, the obtained thermal response is analyzed by different processing algorithms to get the detail visualization.

3- Experimentation

Experimentation was conducted on CFRP sample to validate the proposed RPT technique. Fig. 2 represents the layout of CFRP sample used for testing.

Experimental CFRP specimen of dimensions 24x24x0.5cm with 36 artificially created circular back-holes with various sizes located at various depths (Fig. 2) is energized with a quadratic modulated chirped stimulus for a 0.01-0.1Hz band of low frequencies (Fig. 1) for duration of 100s. The experimental setup shown in Fig. 3 is explained as follows: the experimental side of the object to be tested is placed opposite to the IR camera, the input excitation is applied over the object with a set of halogen lamps placed opposite to the test object [11] and the entire set up is controlled by using a control unit, an IR camera is used to capture the thermal perturbations of the test object simultaneously and later exported to processing unit for further subsurface analysis.

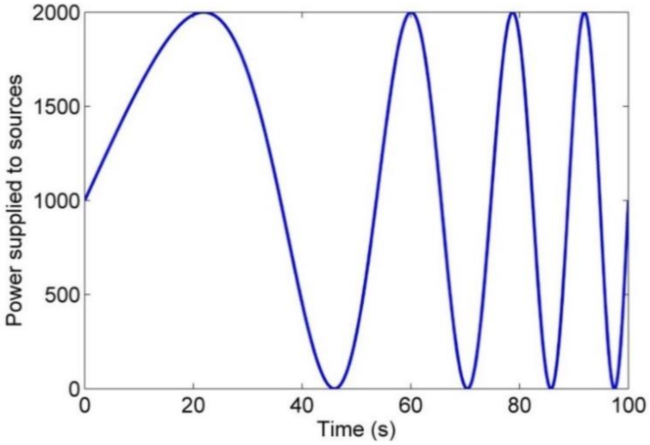


Fig. 1 Schematic of incident quadratic frequency-modulated heat flux of 2kw power at 0.01-0.1Hz frequency

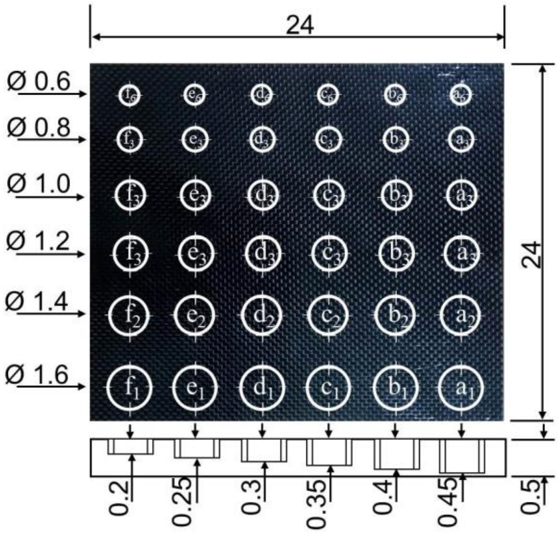


Fig. 2 CFRP specimen layout (dimensions in cm)

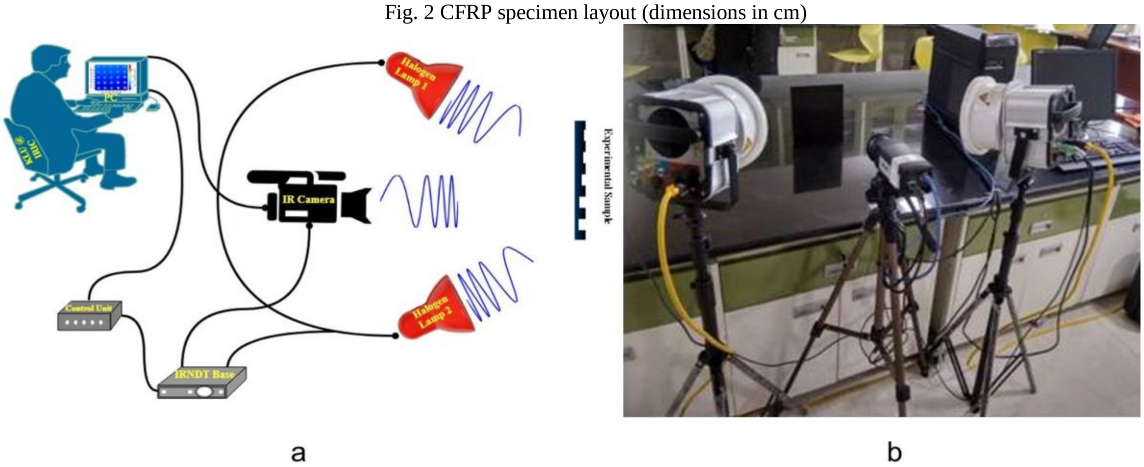


Fig. 3 a. Schematic view and b. Experimental set up of QFMTWI

4- Results and Discussions

The captured thermal response using IR camera consists of both static, dynamic responses [12] by considering DC in the input excitation (Fig. 1). The static response doesn't contain any information regarding subsurface details; it must be pre-processed before processing through various processing algorithms [13]. Hence the static response was removed by using a suitable linear fitting algorithm; the mean removed dynamic response is processed using FFT, PC and RPT techniques to get detailed visualization of subsurface layers is shown in Fig. 4.

4-1- FFT

FFT is the widely used conventional post-processing technique [14, 15] for defect characterization in NSTWI. In this, FFT is applied over each profile as in Eq. (20) of mean removed dynamic thermal response and corresponding phase value is extracted using Eq. (21) and placed in their respective locations [16] to construct the phase grams.

$$F\left(w\right)=\sum_{n=0}^{N-1} f\left(n\right)e^{jwn} \tag{20}$$

$$\phi_{FFT}=\tan^{-1}\left(\frac{\mathrm{Im}\left(F\left(nw\right)\right)}{\mathrm{Re}\left(F\left(nw\right)\right)}\right) \tag{21}$$

Further phase contrast is calculated to characterize [17] the defective and non-defective regions.

4-2- PC

PC is the correlation based matched filter technique, which cross-correlates the referenced non-defective thermal profile with each profile of the mean removed thermal response [18, 19, 20] using Eq. (22). The corresponding normalized correlation coefficients are placed accordingly to construct correlation images.

$$g(\tau)=\int_{-\infty}^{+\infty} s(t)h(t+\tau)dt \qquad (22)$$

Normalized correlation contrast discriminates defective and non-defective regions. Normalized peak delays between the reference correlation profile with defective profiles are used [21] to characterize the subsurface details. Fig. 5 shows the normalized correlation peak delays of reference compressed profile and defective compressed profiles of 1.6cm diameter (defects a_1 to f_1) and corresponding peak delay vs. depth plot. The decreased normalized correlation peak delay value from deeper defect (f_1) to shallowest defect (a_1) discriminates the depth of the defects.

4-3- RPT

In RPT, the orthonormal features of the random vectors are extracted by using the Gram-Schmidt algorithm and later the mean removed dynamic thermal profiles are projected on these vectors to get the normalized random projection coefficients [22]. The random vectors in the orthonormal basis are generated by extracting Eq. (23)

$$V_1=\frac{f_1[n]}{\left|f_1[n]\right\|};V_2=f_2[n]-\left(V_1^T f_2[n]\right)V_1 \qquad (23)$$

The entire procedure is repeated for each random vector and then mean removed dynamic thermal profiles are projected on them. In this, the three-dimensional mean removed dynamic thermal data is converted into two-dimensional data and later this two-dimensional dynamic data is projected on the random vectors generated by using the Gram-Schmidt algorithm [23]. Further, the random projected coefficients are used to discriminate the anomalies and rearranged to get the random projected images.

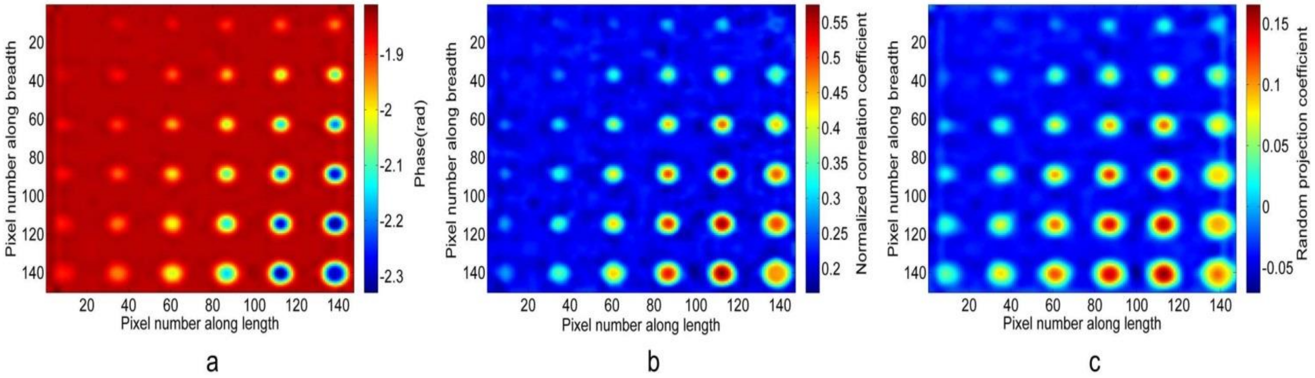


Fig. 4 Processed results of CFRP sample for QFMTWI a. FFT phase at 0.05Hz, b. Correlation image at 7.8s and c. Random projection image at 4th component

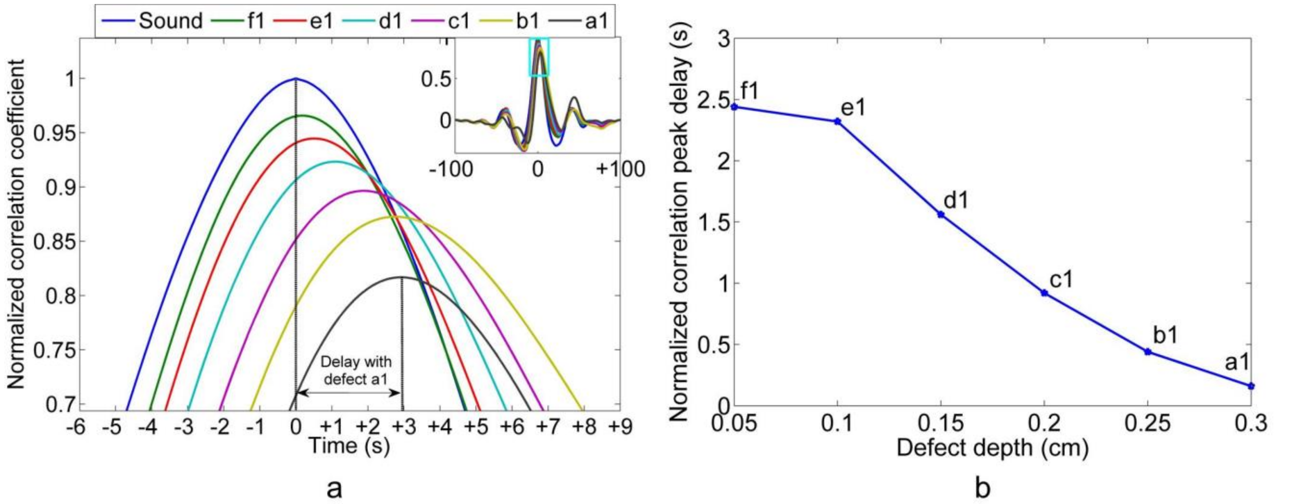


Fig. 5 a. Enlarged (of inset) normalized correlation peak delays of reference and defective profiles of 1.6cm diameter and b. Corresponding peak delay vs. depth of the defect

4-4- SNR

Defect detectability of the processing method is explained by estimating SNR [24, 25] using Eq. (24)

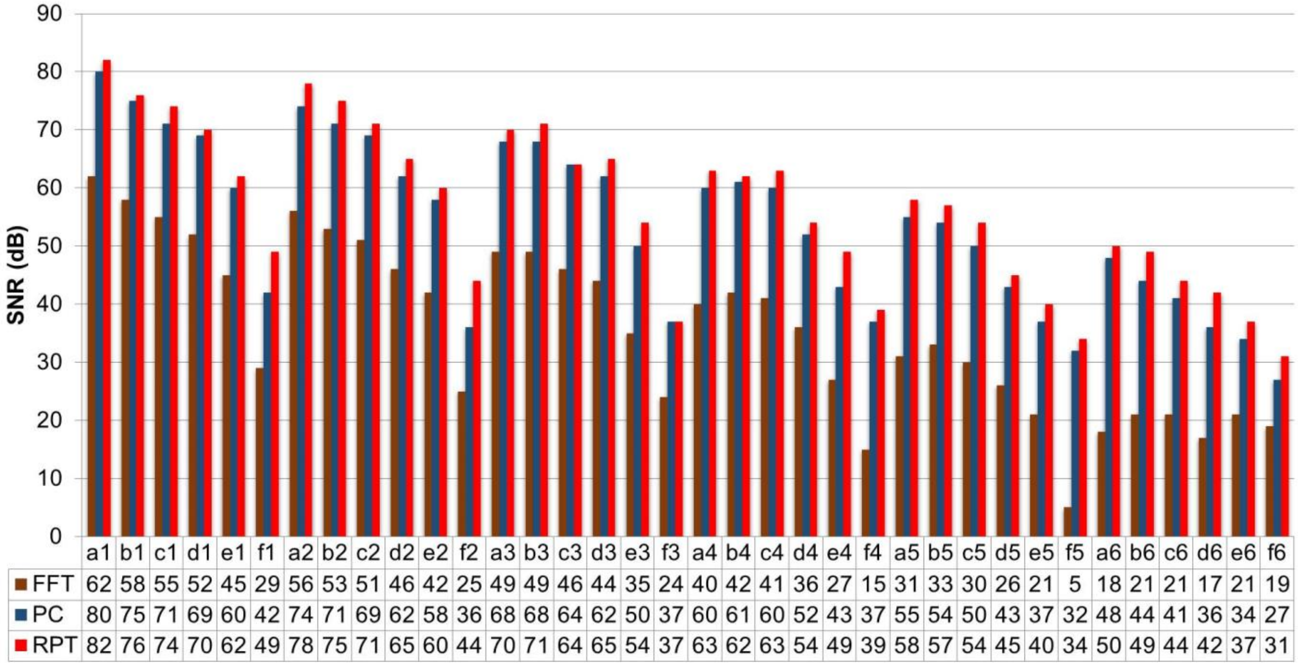
$$SNR = 20\log\left(\frac{\mu_{defective} - \mu_{non-defective}}{\sigma_{non-defective}}\right) \qquad (24)$$

Fig. 6 compares the SNR of each defect of the RPT approach with the conventional FFT and PC based approaches. From Fig. 6, it shows that the RPT based post-processing approach detects the defects better than conventional approaches in terms of SNR.

4-5- FWHM

The qualitative subsurface analysis is done by estimating the size of defects, for this a widely used thermographic performance metric [26] FWHM is computed and estimated defect diameters are shown in the following table 1. Table 1 compares the actual diameter of the defects with estimated FWHM for processed results and its corresponding error values. RPT based post-processing approach gives the nearest diameters of the defects than the conventional approaches.

The result analysis authenticates, that the proposed RPT based statistical approach through 3D green's function analytical analysis detects the deeper and smaller defects with improved sensitivity and resolution by making use of randomly generated orthonormal vectors to discriminate the defective area with sound area and it is validated experimentally for CFRP sample using QFMTWI. To support this SNR and FWHM are calculated and compared with conventional signal processing approaches.



	a1	b1	c1	d1	e1	f1	a2	b2	c2	d2	e2	f2	a3	b3	c3	d3	e3	f3	a4	b4	c4	d4	e4	f4	a5	b5	c5	d5	e5	f5	a6	b6	c6	d6	e6	f6
■ FFT	62	58	55	52	45	29	56	53	51	46	42	25	49	49	46	44	35	24	40	42	41	36	27	15	31	33	30	26	21	5	18	21	21	17	21	19
■ PC	80	75	71	69	60	42	74	71	69	62	58	36	68	68	64	62	50	37	60	61	60	52	43	37	55	54	50	43	37	32	48	44	41	36	34	27
■ RPT	82	76	74	70	62	49	78	75	71	65	60	44	70	71	64	65	54	37	63	62	63	54	49	39	58	57	54	45	40	34	50	49	44	42	37	31

Fig. 6 Comparison of SNR of defects for processed results of FFT, PC and RPT

5- Conclusions

The defect detection capability of QFMTWI through enhanced depth resolution was explored by experimenting with CFRP sample. The projection of randomly generated orthonormal vectors on the thermal response discriminates the deeper and smaller defects with fine contrast. RPT based post-processing technique is validated by comparing the processed results of RPT with conventional FFT and PC based post-processing approaches in terms of SNR and FWHM. The obtained results are evident for the estimation of defect sizing by applying full width at half maxima with an error percentage of 0.99% for deeper and smaller defects. SNR is also considered to further validate the potentiality of detecting the defects and it is measured as minimum of 31dB. Based on the above validation parameters the analysis dependent on RPT technique holds by experimentation is achieved with enhanced detectability, estimation of defect size & evaluation of the subsurface anomalies qualitatively by employing 3D green's function based analytical approach using QFMTWI.

Acknowledgments

This work was partially supported by Naval Research Board, India grant no: NRB-423/MAT/18-19 and FIST sponsored ECE Department under grant no. SR/FST/ET-II/2019/450.

Table 1: Comparison of FWHM of defects of the last column for processed results of FFT phase, PC and RPT

Defect	*Actual defect size (cm)*	*Estimated size (cm)*					
		FFT phase	*% of error*	*PC*	*% of error*	*RPT*	*% of error*
a_1	**1.6**	**1.47**	**8.84**	**1.48**	**8.11**	**1.605**	**0.31**
a_2	**1.4**	**1.45**	**3.45**	**1.43**	**2.10**	**1.41**	**0.71**
a_3	**1.2**	**1.09**	**10.09**	**1.14**	**5.26**	**1.195**	**0.42**
a_4	**1.0**	**0.95**	**5.26**	**0.96**	**4.17**	**1.01**	**0.99**
a_5	**0.8**	**0.71**	**12.68**	**0.73**	**9.59**	**0.805**	**0.62**
a_6	**0.6**	**0.46**	**30.43**	**0.51**	**17.65**	**0.596**	**0.67**

References

[1] X. Maldague, Theory and practice of infrared technology for nondestructive testing, New York: Wiley, 2001.

[2] H. Benítez, C. Ibarra-Castanedo, A. Bendada, X. Maldague, H. Loaiza, and E. Caicedo, "Definition of a new thermal contrast and pulse correction for defect quantification in pulsed thermography", Infrared Physics & Technology, Vol. 51, No. 3, 2008, pp. 160-167.

[3] A. Castelo, A. Mendioroz, R. Celorrio, and A. Salazar, "Optimizing the Inversion Protocol to Determine the Geometry of Vertical Cracks from Lock-in Vibrothermography", Journal of Nondestructive Evaluation, Vol. 36, No. 1, 2016.

[4] C. Ibarra-Castanedo, N. Avdelidis, and X. Maldague, "Quantitative pulsed phase thermography applied to steel plates", Thermosense XXVII, 2005.

[5] R. Mulaveesala, and S. Tuli, "Theory of frequency modulated thermal wave imaging for nondestructive subsurface defect detection", Applied Physics Letters, Vol. 89, No. 19, 2006, p. 191913.

[6] G. V. Subbarao, and R. Mulaveesala, "Quadratic Frequency Modulated Thermal Wave Imaging for Non-Destructive Testing", Progress In Electromagnetics Research M, Vol. 26, 2012, pp. 11-22.

[7] B. Suresh, S. Subhani, A. Vijayalakshmi, V. Vardhan, and V. S. Ghali, "Chirp Z transform based enhanced frequency resolution for depth resolvable non stationary thermal wave imaging", Review of Scientific Instruments, Vol. 88, No. 1, 2017, p. 014901.

[8] A. Sharma, R. Mulaveesala, and V. Arora, "Novel Analytical Approach for Estimation of Thermal Diffusivity and Effusivity for Detection of Osteoporosis", IEEE Sens. J., Vol.20, No. 11, 2020, pp. 6046–6054.

[9] V. S. Ghali, S. Panda, and R. Mulaveesala, "Barker coded thermal wave imaging for defect detection in carbon fibre-reinforced plastics", Insight - Non-Destructive Testing and Condition Monitoring, Vol. 53, No. 11, 2011, pp. 621-624.

[10] B. Suresh, S. Subhani, V. S. Ghali, and R. Mulaveesala, "Subsurface detail fusion for anomaly detection in non-stationary thermal wave imaging", Insight - Non-Destructive Testing and Condition Monitoring, Vol. 59, No. 10, 2017, pp. 553-558.

[11] G.V.P. Chandra Sekhar Yadav, V. S. Ghali, B. Sonali Reddy, B. Omprakash, and Ch. Chaitanya Reddy, "Greens Function Based Analytical Model for Enhanced Defect Detection Using Depth Resolvable Non-Stationary Thermal Wave Imaging", Journal of Green Engineering, Vol. 10, No. 12, 2020, pp. 12933-12947.

[12] S. Subhani, and V. S. Ghali, "Measurement of thermal diffusivity of fiber reinforced polymers using quadratic frequency modulated thermal wave imaging", Infrared Physics & Technology, Vol. 99, 2019, pp. 187-192.

[13] S. Subhani, B. Suresh, and V. S. Ghali, “Quantitative subsurface analysis using frequency modulated thermal wave imaging”, Infrared Physics & Technology, Vol. 88, 2018, pp. 41-47.

[14] M. Pasha, B. Suresh, K. Rajesh Babu, S. Subhani, and G. V. Subbarao, “Barker coded modulated thermal wave imaging for defect detection of glass fiber reinforced plastic”, ARPN Journal of Engineering and Applied Sciences, Vol. 13, No. 10, 2018, pp. 3475-3480.

[15] G.V.P. Chandra Sekhar Yadav, V. Ghali, and N. Baloji, "A Time Frequency-Based Approach for Defect Detection in Composites Using Nonstationary Thermal Wave Imaging", Russian Journal of Nondestructive Testing, Vol. 57, No. 6, 2021, pp. 486-499.

[16] M. Pasha, G. V. Subbarao, B. Suresh, and S. Tabassum, “Inspection of Defects in CFRP based on Principal Components”, International Journal of Recent Technology and Engineering, Vol. 8, No. 3, 2019, pp. 2367-2370.

[17] V. S. Ghali, B. Suresh, and A. Hemanth, “Data Fusion for Enhanced Defect Detectability in Non-Stationary Thermal Wave Imaging”, IEEE Sensors Journal, Vol. 15, No. 12, 2015, pp. 6761-6762.

[18] B. Suresh, J. Sai kiran, and G. V. Subbarao, “Automatic detection of subsurface anomalies using non-linear chirped thermography”, International Journal of Innovative Technology and Exploring Engineering, Vol. 8, No. 6, 2019, pp. 1247-1249.

[19] M. Parvez M, J. Shanmugam, and V. S. Ghali, “Decision tree-based subsurface analysis using Barker coded thermal wave imaging”, Infrared Physics & Technology, Vol. 109, 2020, p. 103380.

[20] A. Vijaya Lakshmi, V. Gopitilak, M. Parvez, S. Subhani, and V. S. Ghali, “Artificial neural networks based quantitative evaluation of subsurface anomalies in quadratic frequency modulated thermal wave imaging”, Infrared Physics & Technology, Vol. 97, 2019, pp. 108-115.

[21] S. Subhani, G. V. P. Chandra Sekhar Yadav, and V. S. Ghali, “Defect characterisation using pulse compression-based quadratic frequency modulated thermal wave imaging”, IET Science, Measurement & Technology, Vol. 14, No. 2, 2020, pp. 165-172.

[22] B. Suresh, M. Manorama, M. Bhupesh, K. Sai Kiran, G. V. P. Chandra Sekhar Yadav, and V. S. Ghali, “Advanced Signal Processing Approaches for Quadratic Frequency Modulated Thermal Wave Imaging”, International Journal of Emerging Trends in Engineering Research, Vol. 7, No. 11, 2019, pp. 599-603.

[23] S. Subhani, B. Suresh, and V. S. Ghali, “Orthonormal projection approach for depth-resolvable subsurface analysis in non-stationary thermal wave imaging”, Insight - Non-Destructive Testing and Condition Monitoring, Vol. 58, No. 1, 2016, pp. 42-45.

[24] A. Vijaya Lakshmi, K. Nagendra Babu, M. Sree Ram Deepak, A. Sai Kumar, G. V. P. Chandra Sekhar Yadav, V. Gopi Tilak, and V. S. Ghali, “A Machine Learning based Approach for Defect Detection and Characterization in Non-Linear Frequency Modulated Thermal Wave Imaging”, International Journal of Emerging Trends in Engineering Research, Vol. 7, No. 11, 2019, pp. 517-522.

[25] A. Vijaya Lakshmi, V. S. Ghali, M. Parvez, G. V. P. Chandra Sekhar Yadav, and V. Gopi Tilak, “Fuzzy C-Means Clustering Based Anomalies Detection in Quadratic Frequency Modulated Thermal Wave Imaging”, International Journal of Recent Technology and Engineering, Vol. 8, No. 3, 2019, pp. 4047-4051.

[26] A. Vijaya Lakshmi, V. S. Ghali, S. Subhani, and N. Baloji, “Automated quantitative subsurface evaluation of fiber reinforced polymers”, Infrared Physics & Technology, Vol. 110, 2020, p. 103456.

Journal of Information Systems and Telecommunication
Vol.10, No.2, April-June 2022, 141-150

http://jist.acecr.org
ISSN 2322-1437 / ESSN:2345-2773

Nonlinear Regression Model Based on Fractional Bee Colony Algorithm for Loan Time Series

Farid Ahmadi[1*], Mohammad Pourmahmood Aghababa[2], Hashem Kalbkhani[2]

[1].Computer Engineering Department, Urmia University of Technology, Urmia, Iran
[2].Faculty Electrical Engineering, Urmia University of Technology, Urmia, Iran

Received: 26 Apr 2021/ Revised: 16 Oct 2021/ Accepted: 15 Jan 2022
DOI:

Abstract

High levels of nonperforming loans provide negative impacts on the growth rate of gross domestic product. Therefore, predicting the occurrence of nonperforming loans is a vital issue for the financial sector and governments. In this paper, an intelligent nonlinear model is proposed for describing the behavior of nonperforming loans. In order to find the optimal parameters of the model, a new fractional bee colony algorithm (BCA) based on fractional calculus techniques is proposed. The inputs of the nonlinear model are the loan type, approved amount, refund amount, and economic realm. The output of the regression model is that whether the current information is for a nonperforming loan or not. Consequently, the model is modified to detect the status of a loan. So, the modified model predicts the occurrence of a nonperforming loan and determines the loan status, i.e., current, overdue, and nonperforming. The proposed procedure is applied to data gathered from an economic institution in Iran. The findings of this study are helpful for the managers of banks, and financial sectors to forecast the future of the loans and, therefore, manage the budget for the upcoming loan requests.

Keywords: Artificial Bee Colony; Fractional Calculus; Nonlinear Economic Model; Loan Status Prediction.

1- Introduction

1-1- Motivation

Banks and financial institutions have played a significant role in balancing the financial status of the people in recent decades owing to the development of the countries and the development of new financing opportunities for the merchants. Lending is one of the primary and popular approaches in financial organizations aiming to loan somebody if the amount borrowed is returned, usually with an interest fee. However, in some cases, the problem of nonperforming loans (NPL) occurs when the borrowed money is not returned in the scheduled period. High levels of NPL mean reducing the income of the banks, which in turn leads to their severe economic losses. Therefore, governments have paid more attention to this issue in recent years. Accordingly, a weakening in bank loan services may cause a delay in economic growth and can be a good reason for the economic crisis. It has been argued that NPLs may create economic stagnation and, therefore, deter economic growth and weaken financial efficacy [1]. On the other hand, high levels of nonperforming loans harm the growth rate of gross domestic product [2]. Given the importance of this issue, the governments and financial sectors should encourage researchers to conduct studies on this issue and banks to implement research results in practice.

Most of the studies have attempted to test whether a particular financial or banking attribute influences NPLs quotient or not. In most cases, just some measurable quantities which depend on the underlying and usually unknown dynamics of the NPLs rate are available. Since many factors are affecting the ratio of NPLs, the inherent nature of NPLs becomes more complicated and nondeterministic. In such circumstances, linear estimators that minimize the variance fail to achieve forecasting objectives. Therefore, it is hard to predict and interpret the long-term future of NPLs with limited features. The complex behavior of the NLPs cannot be easily modeled using the common linear or nonlinear statistical approaches such as autoregressive methods. Moreover, in the previous nonlinear neural network models, the designer should set many model parameters correctly. Thus, it would be better to use alternative nonlinear

✉ **Corresponding Author**
Email Address: f.ahmadi@uut.ac.ir

powerful techniques that utilize the NPLs nature's inherent attributes.

1-2- Contributions

With over 300 years of history, Fractional calculus is a generalization of ordinary differentiation and integration to arbitrary non-integer order. Although it has a long history, it has been used in physics and engineering during the past three decades. It has been recognized that many systems in interdisciplinary fields can be elegantly described using fractional-order differential equations. Moreover, fractional calculus has some exciting features, such as memory of all past events and the precise modeling ability of real-world systems. Using the powerful features of fractional calculus can improve the efficiency of the heuristic algorithms such as BCA.

This paper aims to propose a nonlinear polynomial regression model to forecast the NPLs of the main interest-free institution of Iran named Omid Entrepreneurship Fund (OEF). In order to find the optimal parameters of the proposed nonlinear regression model, a novel fractional heuristic method based on BCA is proposed to achieve a high-speed convergence rate with a high ability to escape local optimum traps. Based on the literature review, fractional BCA was not utilized for loan time series modeling to the best of our knowledge. In simulations, the proposed model is implemented for the data gathered from OEF. Numerical simulations show that this model can predict the future of a loan better than the original BCA and multi-layer perceptron neural network (MLPNN).

Contributions of this paper are as follows.

- Introducing nonlinear fractional BCA (FBCA) as a nonlinear polynomial regression model
- Constructing the four-input single-output heuristic model
- Considering loan type (LT), approved amount (AA), refund amount (RA), and economic filed (EF) as inputs of regression model
- Introducing two regression models for loan status prediction
- The output of the first model is whether the current information is for a nonperforming loan or not
- The second proposed intelligent regression model is modified to detect the exact status of a loan as current, overdue, and nonperforming

This paper is organized as follows. In Section II, the nonlinear regression model is presented. Section III deals with the description of the proposed FBCA. In Section IV, some computer simulations are carried out. Finally, concluding remarks are provided in Section V.

2- Related Works

NPL ratios should stand at low or manageable levels before the crisis. To this end, the NPL ratio prediction method based on previous information is necessary. Various studies have attempted to study the relationship between various economic factors and the NPLs to forecast the NPLs. In [1], a heuristic hybrid classification method has been used to predict the banks' nonperforming loans using some macroeconomic and bank-specific features. In [3], a step-wise discrimination algorithm was introduced to find essential factors for building distance discrimination and Bayesian discrimination models to determine whether an NPL has a zero or positive recovery rate. Three-phase mixture models of logistic regression and artificial neural networks have been developed in [4] to create an economic distress warning system appropriate for Taiwan's banking business. The application of a neural network predictor in forecasting loan recovery in Nigerian financial institutions has been reported [5]. In [6], the principal component analysis technique has been adopted for feature selection of Chinese bank loan default prediction models; then, the models have been assessed using the technique for order preference by similarity to ideal solution (TOPSIS). Authors in [7] have utilized machine learning strategies such as ontology, text and data mining, and multi-agent approaches to develop knowledge-determined automatic acquiescence auditing methods for bank loans. Some studies have used other techniques such as nonlinear regression models for loan prediction [8, 9]. In [10], Novikov's theorem has been utilized to model complicated dynamics of noisy credit risk contagion with time-delay, and the Hopf bifurcation and chaotic behaviors have been evaluated. In [11], some numerical approaches have been applied to discover Hopf bifurcation, inverse bifurcation, and chaos phenomena in the credit risk contagion dynamics. Lahmiri [12] has investigated the fractal inherence and chaotic behavior in returns and volatilities of family business companies of Morocco, using Hurst exponent and an autoregressive model. In [13], the phase synchronization method has been introduced for analyzing the chaotic behavior of stock price and index movements in crisis stages. For identifying the quality of similarity measure of financial time series, three techniques, including information categorization approach, reconstructed phase space clustering strategy, and system methodology with squared Euclidean distances, have been used in [14]. In addition to these works, some studies are aiming at predicting the chaotic financial time series. In [15], a self-organizing map (SOM) neural network and a recommender system have been proposed to cluster and predict stock price time series. Authors in [16] have applied empirical mode decomposition (EMD) and phase spacer construction methods combined with extreme learning machines (ELM) for predicting financial exchange rates' time series forecasting. In [17], the recovery rate of NPL for a European country was modeled using several linear regression models, including linear, linear with Lasso, beta, and inflated beta, as well as a two-stage beta mixture model combined with a logistic regression model. These models were used to estimate future recovery rates for improved risk assessment, capital requirement calculations, and bad debt management. The results showed that the two-stage model outperforms the others. A two-stage model including classification tree-based boosting and support vector regression (SVR) was proposed in [18] in order to estimate the loss given default (LGD). The results indicate that incorporating nonlinearity and boosting improve performance. Artificial intelligence is the primary and most applied technique in the literature for predicting the future of the time series [19-23].

Authors in [24] presented an ensemble-based machine learning model for predicting agricultural loans. For the building model, the dataset was gathered from an agricultural bank in Egypt. Variable selections were used to select important features for the classification. Some classifiers such as logistics regression, k-nearest neighbors, support vector machine, decision tree, and Meta-classifier methods were utilized for training and testing the dataset. The empirical impact of deprived sector lending on the nonperforming loans of commercial banks in Nepal was investigated in [25] that utilizes the ordinary least squares regression. The study establishes empirical relation between deprived sector lending and nonperforming loan of banks. A systematic literature review approach and discussion on the ten-year evolution of credit risk research was performed in [26]. It was found that machine learning is being extensively applied in credit risk assessment, where artificial intelligence applications mainly were found, more specifically artificial neural networks. The study of [27] compares Bayesian networks with artificial neural networks for predicting a recovered value in a credit operation. It was found that ANNs are a more efficient tool for predicting credit risk than the naive Bayesian approach. The convolutional neural network was employed in [28] to automatically extract essential cross features and generate cross-feature embedding from structured data, reducing the need to generate hand-crafted cross features. The local logit regression was employed in [29] for defaulted loan recoveries, and it was found that this model is robust to nonlinearity and non-normality of errors. Also, the empirical features of the local logit model were exploited to improve the specification of the standard regression for the fractional response variable model. The credit defaulter dataset of Bangladeshi banks was considered in [30], and several traditional machine learning classifiers were utilized to predict the delinquent clients who possessed the highest probability of short-term credit recovery. The different models for predicting the recovery rate on borrower level, including linear and quantile regressions, decision trees, neural networks, and mixture regression models, were compared in [31]. Authors in [32] proposed a system for building accurate models using interpretable state-of-the-art ML algorithms and explainable artificial intelligence techniques to explain individual instances for supporting business decisions. The basic user information and added the user's consumption features were considered in [33] to construct a loan risk assessment model that integrates features. Consumption features were extracted from two aspects: portrait features and sequence features. Finally, the features are combined and fed into the fully connected layer, and the probability of default loan is calculated.

Heuristic algorithms are higher-level procedures that are designed to find an excellent solution to optimization problems. Over the last decades, many intelligent algorithms have been developed to solve engineering optimization problems and time series forecasting [34-37]. Most of the developed algorithms are based on linear or nonlinear programming approaches. However, some complex problems cannot be solved with linear or nonlinear programming ways. For example, if the problem has more than one locally optimal solution, these methods must start with different initial points. Heuristic alternatives can find optimal solutions in complex problems using their capabilities (combination of randomness and rules, high speed, etc.). Bee colony algorithm (BCA) is one of the heuristic optimization methods introduced in 2005 by Dervis Karaboga [38]. The intelligent behavior of honey bees inspires the basic idea of BCA. Such as other swarm optimization techniques, BCA has a population-based search strategy in which bees fly around in a multidimensional search space, and the employed and onlooker bees choose food sources depending on the experience of themselves and their nestmates adjust their positions. Similar to other numerical optimization algorithms, BCA faces some challenges. For example, BCA shows a slow convergence speed during the search process.

Moreover, BCA quickly falls into local minima when handling complex multimodal problems [39]. Therefore, some mechanisms are required to improve the quality of the original BCA. In order to improve the accuracy and efficacy of the BCA, this paper proposes an innovative fractional calculus-based method.

3- Fractional Calculus and Nonlinear Regression Model

In this section, first, some preliminaries for the fractional calculus are provided. Then, the proposed nonlinear regression model is introduced.

3-1- Fractional Calculus

Definition 1. The fractional integration of order of function $f(t)$ in the sense of Riemann-Liouville is as follows [40]:

$$ {}_{t_0}I_t^{\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_{t_0}^{t} \frac{f(\tau)}{(t-\tau)^{1-\alpha}} d\tau \qquad (1)$$

where t_0 is the initial time and the Euler's Gamma function $\Gamma(\alpha)$ is defined as:

$$\Gamma(\alpha) = \int_0^{\infty} z^{\alpha-1} e^{-z} dz \qquad (2)$$

Definition 2. The Riemann-Liouville fractional derivative of the order $\alpha \in$ ¡ of function $f(t)$ is given by [40]:

$$\begin{aligned} {}_{t_0}^{RL}D_t^{\alpha} f(t) &= \frac{d^{\alpha} f(t)}{dt^{\alpha}} \\ &= \frac{1}{\Gamma(m-\alpha)} \frac{d^m}{dt^m} \int_{t_0}^{t} \frac{f(\tau)}{(t-\tau)^{\alpha-m+1}} d\tau \end{aligned} \qquad (3)$$

where $m-1 < \alpha \le m,\ m \in N$.

Definition 3. Grunwald–Letnikov fractional derivative is defined as [40]:

$$ {}_{t_0}^{GL}D_t^{\alpha} f(t) = \lim_{h \to 0} \frac{1}{h^{\alpha}} \sum_{j=0}^{\left[\frac{t-t_0}{h}\right]} (-1)^j \binom{\alpha}{j} f(t-jh) \qquad (4)$$

where $[.]$ is the integer part operator and $\binom{\alpha}{j}$ is the fractional binomial coefficient defined by:

$$\binom{\alpha}{j} = \begin{cases} 1 & j = 0 \\ \dfrac{\alpha(\alpha-1)...(\alpha-j+1)}{j!} & j > 0 \end{cases} \qquad (5)$$

Remark 1. Eq. (3) reveals that although the integer-order operators are represented by finite series, the fractional-order counterparts are given via infinite series. This indicates that integer-order derivatives are local operators in contrast to the fractional operators that have, implicitly, a memory of all past events. So, it can be concluded that the fractional-order systems have limitless memory (infinite-dimensional) while the integer-order systems posse limited memory (they are finite-dimensional).

Remark 2. A discretization scheme using a finite difference equation should apply fractional-order derivatives and integrals in real applications. The well-known and most straightforward technique is to benefit memory length expansion from (4) to construct a direct discretization. This methodology relies on Grunwald-Letnikov, and Riemann-Liouville's definitions are equivalent in reasonable conditions for a broad class of functions [40]. Accordingly, the following definition of the approximated fractional difference is adopted in this article:

$$\Delta_h^{\alpha} x(t) = \frac{1}{h^{\alpha}} \sum_{j=0}^{k} (-1)^j \binom{\alpha}{j} x(t-jh) \qquad (6)$$

where $k \in$ ¥ shows the number of samples for which the approximation of the derivative is computed.

3-2- Nonlinear Regression Model

In this paper, we propose a nonlinear regression model (NRM) based on FBCA to predict the status of the loans. The nonlinear regression attempts to find the functional relationship between the inputs and outputs. Here, the coefficients of the nonlinear regression are estimated by the FBCA. The proposed FBCA is designed to minimize the mean square error (MSE) between the predicted and target outputs. In this paper, the following nonlinear polynomial regression model is used:

$$F_t = K + \sum_{i=1}^{n} k_i x_i + \sum_{i=1}^{n} q_i x_i^{r_i} + \sum_{i=1}^{n} \sum_{j=i+1}^{n-1} p_{i,j} x_i x_j \qquad (7)$$

where K , k_i , q_i , r_i , and p_{ij} are constant regression parameters and F denotes the loan status.

Remark 3. It should be noted that FBCA determines the fair values for the constant parameters of the model.

Remark 4. The proposed regression model involves two phases; train and test. The training data is applied to the model in the training phase, and the output error is computed. The model parameters are computed using FBCA to minimize this error. After *achieving a less enough train error*, the model parameters are fixed. The test phase is performed on unseen data to evaluate the model's generalization ability for the new data.

Remark 5. The mean square error (MSE) is calculated as:

$$MSE = \frac{\sum_{t=1}^{N} (F_t - Y_t)^2}{N} \qquad (8)$$

where Y_t represents the desired output and N denotes the number of data.

4- The Proposed FBCA Method

In this section, first, the original BCA is restated. Then, the proposed FBCA is introduced.

4-1- The Original BCA

As an intelligent optimization tool, BCA provides a population-based search scheme such that the artificial bees modify individuals' called foods positions with time.

In this process, the bee's goal is to look for food sources with high nectar amounts and, at last, the one with the highest nectar. In the ABC system, artificial bees fly around in a multidimensional search space, and some (employed and onlooker bees) select food sources based on their experience and their nestmates and regulate the corresponding positions. Some (scouts) fly and select the food sources in a stochastic manner with no experience. The bees will memorize the new position and forget the previous one if the nectar amount of a new source is higher than that in their memory. Hence, the BCA system combines local search techniques, performed by employed bees and onlooker ones, with global search approaches modeled by onlookers and scouts. This process tries to balance exploration and exploitation.

The first half of the population comprises employed bees in the BCA scheme, and the second half composes the onlooker bees. The number of employed bees or the onlooker bees is set as the number of solutions in the population [26].

The BCA produces a stochastically distributed initial swarm of N_{pop} solutions. Assume $\mathbf{x}_i = \{x_{i1}, x_{i2}, ..., x_{id}\}$ represents the *i*th solution in the swarm, where *d* is the dimension size. Each employed bee $\mathbf{x}_i$ produces a new potential solution v_i in the neighborhood of its current position as:

$$v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}) \qquad (9)$$

where x_{kj} is a randomly selected candidate solution $(k \neq i)$, j stands for a random dimension index selected from the set $\{1, 2, ..., d\}$, and φ_{ij} represents a random number within [-1, 1]. Once the new candidate solution v_i is built, a greedy selection is adopted. If the fitness value v_i is higher than that of its parent v_i, then update $\mathbf{x}_i$ with v_i; otherwise $\mathbf{x}_i$ is held unchangeable.

After all employed bees complete the search process, they share the information about their food sources with the onlooker bees through waggle dances. An onlooker bee assesses the nectar information adopted from all employed bees and chooses a food source with a probability related to its nectar amount. This probabilistic selection is equal to a roulette wheel selection process which is given as:

$$p_i = \frac{\gamma_i}{\sum_{j=1}^{N_{pop}} \gamma_j} \qquad (10)$$

where γ_i is the fitness value of the *i*th solution in the population. It is clear that the better *i*th solution, the higher the probability of the *i*th food source being chosen. The food source is abandoned if a position cannot be improved over a predefined number of cycles. Letting the abandoned source as $\mathbf{x}_i$, the scout bee finds a new food source to be exchanged with $\mathbf{x}_i$ as:

$$x_{ij} = b_j + \beta(u_j - b_j) \qquad (11)$$

where β is a random number uniformly distributed within the range [0, 1] and b_j and u_j indicate the lower and upper boundaries of the *j*th dimension, respectively.

4-2- The FBCA Algorithm

Globally optimizing a fitness function in a given search domain finds its global optima fast and without sticking in local optima. Slow convergence of BCA before achieving an exact solution is a drawback related to its lack of appropriate memory in the bee position update (9). One can rewrite (9) as follows:

$$v_{ij} - x_{ij} = \varphi_{ij}(x_{ij} - x_{kj}) \qquad (12)$$

It is clear from $v_{ij} = x_{ij}(t+1)$, where *t* stands for the time or iteration, and the left side of (12) $v_{ij} - x_{ij}$ is the discrete version of the fractional derivative. In other words, the left side of (12) represents the fractional difference (6) with order $\alpha = 1$ and $h = 1$, i.e.,

$$\Delta_1^1 v_{ij} = \varphi_{ij}(x_{ij} - x_{kj}) \qquad (13)$$

Also, using a real order difference (i.e., fractional difference) can result in a more generalized position updating relation. So, generalization of the fractional difference order to a real number $0 \le \alpha \le 1$ may access a smoother variation and a longer memory effect. Thus, considering the first $k = 4$ terms of the fractional difference given by (6), the following fractional position updating relation is proposed:

$$\begin{aligned} &v_{ij} - \alpha x_{ij}(t) - \frac{1}{2}\alpha x_{ij}(t-1) - \frac{1}{6}\alpha(1-\alpha)x_{ij}(t-2) \\ &-\frac{1}{24}\alpha(1-\alpha)(2-\alpha)x_{ij}(t-3) = \varphi_{ij}(x_{ij} - x_{kj}) \end{aligned} \qquad (14)$$

or

$$\begin{aligned} &v_{ij} = \alpha x_{ij}(t) + \frac{1}{2}\alpha x_{ij}(t-1) + \frac{1}{6}\alpha(1-\alpha)x_{ij}(t-2) \\ &+\frac{1}{24}\alpha(1-\alpha)(2-\alpha)x_{ij}(t-3) + \varphi_{ij}(x_{ij} - x_{kj}) \end{aligned} \qquad (15)$$

4-3- Loan Status Prediction by FBCA-Based Nonlinear Regression

In this study, the nonlinear regression based on modified BCA is used to forecast the loans' future status. As mentioned before, nonlinear regression is a regression analysis that uses a combination of the independent variables to solve the nonlinear problems. Most of the studies have been used optimization algorithms and neural networks to achieve the best weights for the NRM. In this study, the introduced FBCA search is used to estimate the best weights for the NRM. Some essential features of a loan are considered as the inputs of the regression model to make a prediction. In this study, the inputs are selected as the features loan type (LT), approved amount (AA), refund amount (RA), and economic filed (EF), which are available from the data obtained from OEF. LT and EF are qualitative variables, while AA and RA are quantitative variables. The inputs LT, AA, and RA are the loan features, while EF is not dependent on the loan and is related to the scope of application; hence, the proposed loan status prediction can be considered multimodal.

Two structures are chosen for the output of the model. In the first structure, we are attempting to detect that whether a loan with a known feature vector is related to an NPL or not. In this case, the model has a single output. The second structure is proposed for classification (and therefore for prediction) of the loans' status, including current loan (CL), overdue loan (OL), and NPL. In this case, the model has several outputs. In both cases, the model estimates the loan status only based on its corresponding input vector. However, the main issue is selecting optimal parameters (weights) for the model. Based upon previous discussions, the proposed FBCA is applied to find the optimal weights while minimizing the MSE (or maximizing inverse of MSE). After finding the suitable weights, the model structure is fixed, and the model's generalization capability is examined in the test mode using unseen data. The main procedure of the proposed scheme for loan status prediction is given by the pseudo-codes provided in Algorithms 1 and 2.

5- Data and Numerical Analysis

In this section, first, a brief description of the adopted data is given. Then, the efficiency of the proposed FBCA is tested using a set of benchmark functions. Also, comprehensive numerical simulation and analysis are provided to verify the data's complex behavior and forecast the future status of the loans.

5-1- Financial Data

Recently, some microfinance institutes have the mission of providing limited loans with no (or at least a minimum) interest. The target population of such interest-free organizations is low-income people who lack access to the financial services of other banks or traditional financial institutions. The essential condition for requesting an interest-free loan is that the recipient must prove setting up a small-scale enterprise. The essential goal of such microfinance institutes is to help low-income people get better access to financial services and finance small or medium projects. OEF is the most essential and essential interest-free institution of Iran. OEF has at least one branch in each province of Iran. The five-year loan features of two branches of OEF with the highest level of activity in the considered time horizon have been chosen and call them B_1 and B_2 branches. Since an NPL is defined as a loan with no payment for at least 18 months for the OEF policies, the last two years' data (which do not contain an NLP) are removed. The major statistical attributes for the numerical features of the original input data are shown in Table 1. It is noted that in all simulations, all data is normalized to the range [-1, 1].

Algorithm 1: Train phase of the proposed FBCA

```
Step 1. Choose the inputs (loan features) and outputs (loan status) of the model.
Step 2. Initialize the parameters of the model (such as maximum iteration, population size, etc.).
Step 3. Set the weights of the regression model as the decision variables.
Step 4. Randomly initialize the swarm.
Step 5. For each employed bee, generate a new candidate solution $v_i$ according to (15).
Step 6. Evaluate the fitness of $v_i$ using the regression model (7) and the inverse of MSE (8).
Step 7. Use a roulette wheel selection (10) to choose a better one between $\mathbf{x}_i$ and $v_i$ as the new $\mathbf{x}_i$ .
Step 8. The scout bee determines the abandoned $\mathbf{x}_i$ , if exists and update it by (11).
Step 9. Update the solution pool, and the best solution found so far, and iteration increases one unit.
Step 10. If the number of iterations reaches the maximum value, stop the algorithm and output the results; otherwise, go to Step 5.
End of train phase.
```

Algorithm 2: The test phase of the proposed FBCA

```
Step 1. Fix the parameters (weights) of the model (7) using the best solution found in the Train phase.
Step 2. Apply the test data to the fixed model and obtain the outputs.
Step 3. Compute the error (8).
End of the Test phase.
```

Table 1: Statistical attributes of the input data

Branch	B_1		B_2	
Number of data	1661		1531	
Input feature	AA	RA	AA	RA

Mean	8.40×10^7	68.43	1.42×10^8	54.67
Median	9.0×10^7	80.7800	1.0×10^8	50.01
Min.	5.0×10^6	0	1.0×10^7	0
Max.	1.0×10^9	100	4.0×10^{10}	100
Standard deviation	6.72×10^7	34.59	1.02×10^9	36.11
Variance	4.52×10^{15}	1.1965×10^3	1.05×10^{18}	1.30×10^3

5-2- Testing Efficiency of FBCA

Here, the efficiency of the proposed FBCA is verified using a set of benchmark functions. To avoid any misinterpretation of the optimization results related to the choice of any particular initial parameters, we performed each test 50 times, starting from various randomly selected solutions drawn from the standard uniform distribution on the search domain specified in the usual literature. The results of FBCA tests on eight functions given in Table 2 are provided in Table 3. To assess the effectiveness of the proposed FBCA procedure, the rate of successful minimizations (r_{SM}), the average of the objective function evaluation numbers (n_{obj}), and the average error $(\bar{\varepsilon})$ are taken into account as minimization criteria. The rate of successful minimizations denotes the ratio of the number of trials that FBCA reaches the minimum value to the total number of evaluation trials. Hence, it can be formulated as: $r_{SM} = n_{\min} / n_{total}$, where $n_{\min}$ is the number of trials FBCA reaches the minimum value and n_{total} is the total number of evaluation trials. The objective function evaluation number denotes the number of iterations of FBCA before reaching the objective function to the minimum value or the number of iterations performed before the convergence of FBCA.

Table 3: Results of FBCA for eight benchmark functions

Function	r_{SM} (%)	n_{obj}	$\bar{\varepsilon}$
SH	100	43	0.000
ES	100	39	0.000
S_n	100	86	0.000
NQ_n	99	110	0.001
RA_n	99	115	0.001
SC_n	99	112	0.001
R_n	99	138	0.002
Z_n	99	142	0.002

The performance of FBCA is then compared to the original BCA and GA algorithms. The experimental results obtained for the test functions, using the three different methods, are given in Table 4. In our simulations, each population in GA has 15 chromosomes, and a swarm in BCA has 15 particles. Other parameters of the three algorithms are selected by trial and error. For each function, we give the average number of function evaluations for 100 runs.

Table 4: The average number of objective function evaluations

Function	FBCA	BCA	GA
SH	43	132	224
ES	39	128	201
S_n	86	164	416
NQ_n	110	237	513
RA_n	115	389	650
SC_n	112	411	780
R_n	138	583	869
Z_n	142	691	979

The convergence plots of the three methods for two functions Sn and NQn, are illustrated in Figs. 1-2. One sees that in Fig. 1, the GA is trapped in a local minimum, and the original BCA spends more time finding the global minima.

Table 2: The benchmark test functions

Name	Function	D	Search space	$\mathbf{x}^*$	$f\left(\mathbf{x}^*\right)$
SH	$f_1(\mathbf{x}) = \left(\sum_{j=1}^{5} j\cos((j+1)x_1 + j)\right)\left(\sum_{j=1}^{5} j\cos((j+1)x_2 + j)\right)$	2	$[-10,10]^D$	18 minimum	-186.7
ES	$f_2(\mathbf{x}) = -\cos(x_1)\cos(x_2)\exp(-((x_1-\pi)^2 + (x_2-\pi)^2))$	2	$[-100,100]^D$	$[-\pi, \pi]$	-1
S_n	$f_3(\mathbf{x}) = \sum_{i=1}^{D} x_i^2$	10	$[-100,100]^D$	$[0,0,...,0]$	0
NQ_n	$f_4(\mathbf{x}) = \sum_{i=1}^{D} ix_i^4$	15	$[-10,10]^D$	$[0,0,...,0]$	0
RA_n	$f_5(\mathbf{x}) = \sum_{i=1}^{D}\left(x_i^2 - 10\cos(2\pi x_i) + 10\right)$	20	$[-5,5]^D$	$[0,0,...,0]$	0
SC_n	$f_6(\mathbf{x}) = \sum_{i=1}^{D} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	20	$[-500,500]^D$	$[1,1,...,1]$	$-418.98D$
R_n	$f_7(\mathbf{x}) = -\sum_{i=1}^{D} -\left(100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2\right)$	25	$[-5,10]^D$	$[1,1,...,1]$	0
Z_n	$f_8(\mathbf{x}) = \sum_{i=1}^{D} x_i^2 + \left(\sum_{i=1}^{D} 0.5ix_i\right)^2 + \left(\sum_{i=1}^{D} 0.5ix_i\right)^4$	30	$[-5,10]^D$	$[1,1,...,1]$	0

Also, from Fig. 2, both GA and BCA cannot find the optimum global point. However, the proposed FBCA can achieve the global minimum solution with fewer function evaluation numbers. Therefore, it can be concluded that the FBCA gains more accurate results compared to the original BCA and GA methods. Moreover, FBCA is faster than the GA and BCA methods.

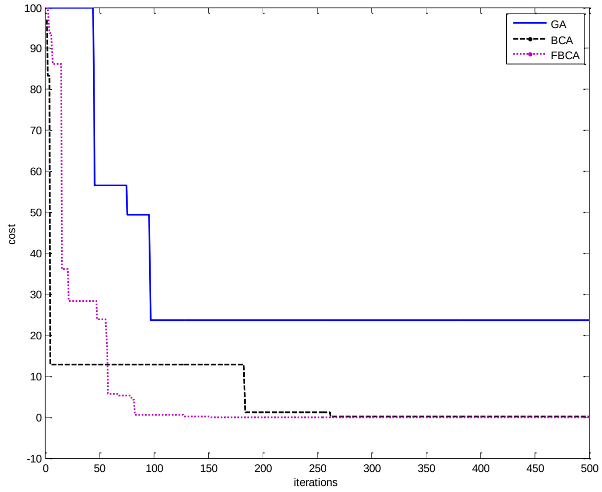


Fig. 1 Typical convergence diagrams of the three methods for S_n function

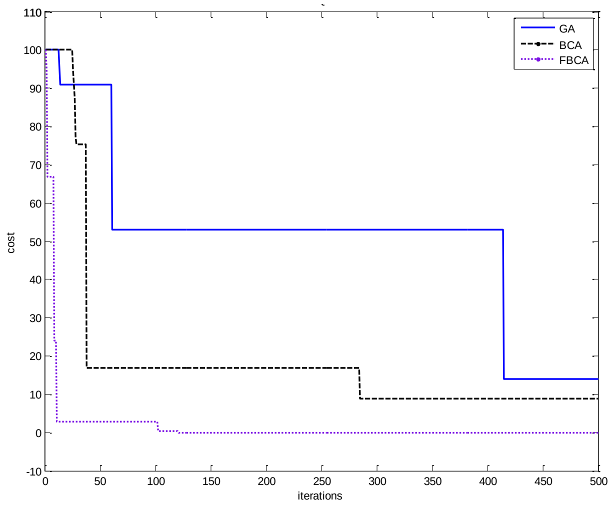


Fig. 2 Typical convergence diagrams of the three methods for NQ_n function

5-3- NPL Prediction

We attempted to forecast the occurrence of NLPs with the heuristic nonlinear prediction method. In this case, a nonlinear regression model was developed using the proposed FBCA. Based on simulations, we assume that if the model's output (7) is in the range (0,100], it denotes an NLP, and if the output is within (100, 200], it does not stand for an NPL. Outranges are missed, and the simulation is repeated for them. Based on the mentioned pseudo-code, the parameters of the model (7) are found. The initial parameters of the optimization algorithm are chosen as follows: $n = 50$, maximum number of iterations = 1000, $p = 0.1$ and $\alpha = 0.05$. To compare the results, a feed-forward MLPNN with one hidden layer of 10 neurons is also implemented to predict the output of the loans. To evaluate the performance of the algorithms, Precision, Recall, and F-measure criteria are adopted as follows:

$$\text{Precision}=\frac{TP}{TP+FP} \tag{16}$$

where TP denotes the number of correctly detected loans as NPLs and FP is incorrectly identified as NPLs. The recall is defined as the ratio of the number of correct loan classes to the sum of the corrected ones with the loans which are misidentified:

$$\text{Recall} = \frac{TP}{TP+FN} \tag{17}$$

where FN is the number of loans that are incorrectly labeled as NLPs. F-measure is also defined as a measure of the test accuracy:

$$\text{F-measure} = 2\,\frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \tag{18}$$

Table 5 compares the results of both methods. It is observed that the proposed nonlinear intelligent model can detect the NLPs with high accuracy compared to the MLPNN. As a result, the predicted values can identify and predict NPLs before they become uncontrollable. Therefore, these predictions can help the managers of financial sectors to develop the necessary policies regarding the budget and the remainder of the loans.

Table 5: Comparative results for NPL detection using the proposed method and MLPNN

Method	**Precision**		**Recall**	
	B_1	B_2	B_1	B_2
Proposed method	0.9712	0.9800	0.9502	0.9471
BCA	0.9031	0.9082	0.9279	0.9206
MLPNN	0.8523	0.8491	0.9042	0.9154
Method	**F-measure**		**Number of function calls**	
	B_1	B_2	B_1	B_2
Proposed method	0.9606	0.9633	570	592
BCA	0.9153	0.9144	713	774
MLPNN	0.8775	0.8810	851	886

5-4- Status Prediction

There are three outputs for forecasting the status of the loans, including current, overdue, and nonperforming. Similar to the previous scenario, we assign a linear portion for each output. In this case, the first output gets the values in the range (0, 100], the second output gets (100, 200], the third output stands for (200, 300], and the final output is assigned to (300, 400]. Outranges are missed, and the simulation is repeated for them. The model's parameters (6) are calculated using the proposed procedure based on the mentioned pseudo-code. The initial parameters of the optimization algorithm are chosen as follows: n = 70, the maximum number of iteration = 1000, p = 0.1 and α = 0.05. To compare the results, a feed-forward MLPNN with one hidden layer of 20 neurons is also implemented to predict the output of the loans. To evaluate the performance of the algorithms, the last five criteria are considered as the comparison criteria. Table 6 shows the results of both techniques. The proposed nonlinear heuristic model outperforms the MLPNN in all criteria. Consequently, the proposed model can identify the status of the loans to help the managers of banks build up the necessary policies for the budget and the loans' customers. Although FBCA has more computational complexity than the BCA, it enhances the accuracy considerably.

Table 6: Comparative results for status prediction using the proposed method and MLPNN

Method	**Precision**		**Recall**	
	B_1	B_2	B_1	B_2
Proposed method	0.9742	0.9782	0.9520	0.9511
BCA	0.9061	0.9021	0.9183	0.9212
MLPNN	0.8751	0.8791	0.9172	0.9090
Method	**F-measure**		**Number of function calls**	
	B_1	B_2	B_1	B_2
Proposed method	0.9630	0.9645	602	627
BCA	0.9122	0.9115	751	808
MLPNN	0.8957	0.8938	884	922

The value of k denotes the number of fractional difference terms, and its value depends on the trade-off between complexity and accuracy. Considering the higher values for k increases both complexity and accuracy and vice-versa. Also, the value of α, which denotes the of fractional difference order, was chosen by trial and error. The lower values of α have smoother results, but reduce the generalization. Table 7 compares the performance of the proposed method for loan status prediction in terms of F-measure for several pairs of (α, k).

Table 7: Performance of the proposed method for different values of pair (α, k)

(α, k)	(0.01, 2)	(0.01, 4)	(0.01, 6)	(0.01, 8)
F-measure	0.9527	0.9719	0.9774	0.9793
(α, k)	(0.03, 2)	(0.03, 4)	(0.03, 6)	(0.03, 8)
F-measure	0.9508	0.9658	0.9689	0.9709
(α, k)	(0.05, 2)	(0.05, 4)	(0.05, 6)	(0.05, 8)
F-measure	0.9481	**0.9645**	0.9672	0.9694
(α, k)	(0.07, 2)	(0.07, 4)	(0.07, 6)	(0.07, 8)
F-measure	0.9472	0.9619	0.9651	0.9638

6- Concluding Remarks

This study provided a new nonlinear regression tool to detect the status of the loans using a limited knowledge about the features affecting the loans. To obtain an optimal model, a new version of BCA is constructed using fractional calculus theory. The effectiveness of the fractional optimization method is verified using test benchmark functions, and it is shown that the new algorithm is fast and reliable. We provide a flexible nonlinear heuristic model that can identify nonperforming loans and any status of the loans. The simulation results using data gathered to form a financial sector in Iran showed that the proposed scheme can effectively forecast the occurrence of NLPs. Moreover, the introduced approach can detect the status of the loan using an average input feature vector. The findings of this study are helpful for the managers of banks and financial sectors to use the nonlinear models to detect the potential nonperforming loans before they become uncontrollable.

Acknowledgment

Omid Entrepreneurship Foundation (OEF) of Iran financially supported this research.

References

[1] Y. Zhang, G. Yu, and D. Yang, "Predicting nonperforming loan of business bank by multiple classifier fusion algorithms," Journal of Interdisciplinary Mathematics, vol. 19, no. 4, pp. 657-667, 2016.

[2] A. Belgrave, G. Kester, and M. Jackman, "Industry Specific Shocks and Nonperforming Loans in Barbados," The Review of Finance and Banking, vol. 4, no. 2, 2012.

[3] Y. Tang, H. Chen, B. Wang, M. Chen, M. Chen, and X. Yang, "Discriminant analysis of zero recovery for China's NPL," Advances in Decision Sciences, vol. 2009, 2009.

[4] S. L. Lin, "A new two-stage hybrid approach of credit risk in banking industry," Expert Systems with Applications, vol. 36, no. 4, pp. 8333-8341, 2009.

[5] A. O. Adewusi, T. B. Oyedokun, and M. O. Bello, "Application of artificial neural network to loan recovery prediction," International Journal of Housing Markets and Analysis, vol. 9, no. 2, pp. 222-238, 2016.

[6] G. Kou, Y. Peng, and C. Lu, "MCDM approach to evaluating bank loan default models," Technological and Economic Development of Economy, vol. 20, no. 2, pp. 292-311, 2014.

[7] P. Saha, I. Bose, and A. Mahanti, "A knowledge based scheme for risk assessment in loan processing by banks," Decision Support Systems, vol. 84, pp. 78-88, 2016.

[8] Z. Bitvai and T. Cohn, "Predicting peer-to-peer loan rates using bayesian nonlinear regression," in Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 2203-2209.

[9] R. Calabrese, "Predicting bank loan recovery rates with a mixed continuous- discrete model," Applied stochastic

models in business and industry, vol. 30, no. 2, pp. 99-114, 2014.

[10] T. Chen, X. Li, and J. He, "Complex dynamics of credit risk contagion with time-delay and correlated noises," in Abstract and Applied Analysis, 2014, vol. 2014: Hindawi.

[11] T. Chen, J. He, and J. Wang, "Bifurcation and chaotic behaviour of credit risk contagion based on Fitzhugh–Nagumo system," International Journal of Bifurcation and Chaos, vol. 23, no. 07, p. 1350117, 2013.

[12] S. Lahmiri, "On fractality and chaos in Moroccan family business stock returns and volatility," Physica A: Statistical Mechanics and its Applications, vol. 473, pp. 29-39, 2017.

[13] S. Radhakrishnan, A. Duvvuru, S. Sultornsanee, and S. Kamarthi, "Phase synchronization based minimum spanning trees for analysis of financial time series with nonlinear correlations," Physica A: Statistical Mechanics and its Applications, vol. 444, pp. 259-270, 2016.

[14] Q. Tian, P. Shang, and G. Feng, "The similarity analysis of financial stocks based on information clustering," Nonlinear Dynamics, vol. 85, no. 4, pp. 2635-2652, 2016.

[15] B. B. Nair, P. S. Kumar, N. Sakthivel, and U. Vipin, "Clustering stock price time series data to generate stock trading recommendations: An empirical study," Expert Systems with Applications, vol. 70, pp. 20-36, 2017.

[16] H.-L. Yang and H.-C. Lin, "Applying the hybrid model of EMD, PSR, and ELM to exchange rates forecasting," Computational Economics, vol. 49, no. 1, pp. 99-116, 2017.

[17] H. Ye and A. Bellotti, "Modelling recovery rates for nonperforming loans," Risks, vol. 7, no. 1, p. 19, 2019.

[18] Y. Tanoue and S. Yamashita, "Loss given default estimation: a two-stage model with classification tree-based boosting and support vector logistic regression," Journal of Risk, vol. 21, no. 4, 2019.

[19] M. Otair and E. Hattab, "An implementation of a new type of online auction," International Journal of E-Business Research (IJEBR), vol. 4, no. 2, pp. 88-102, 2008.

[20] N. S. Sidnal and S. S. Manvi, "Intelligent Agent Based Model for Auction Service Discovery in Mobile E-Commerce," International Journal of E-Business Research (IJEBR), vol. 8, no. 1, pp. 76-97, 2012.

[21] Y. Lu, Z. Zeng, H. Wu, G. G. Chua, and J. Zhang, "An intelligent system for taxi service: Analysis, prediction and visualization," AI Communications, vol. 31, no. 1, pp. 33-46, 2018.

[22] E. Trandafili and M. Biba, "A review of machine learning and data mining approaches for business applications in social networks," International Journal of E-Business Research (IJEBR), vol. 9, no. 1, pp. 36-53, 2013.

[23] R. Chandra and S. Chand, "Evaluation of co-evolutionary neural network architectures for time series prediction with mobile application in finance," Applied Soft Computing, vol. 49, pp. 462-473, 2016.

[24] M. A. Elnaggar, M. A. EL Azeem, and F. A. Maghraby, "Machine Learning Model for Predicting Non-performing Agricultural Loans," in The International Conference on Artificial Intelligence and Computer Vision, 2020: Springer, pp. 395-404.

[25] S. K. Oli, "Deprived Sector Lending and Non-Performing Loans in Nepal," Applied Economics and Finance, vol. 8, no. 4, pp. 1-10, 2021.

[26] F. Medeiros Assef and M. T. Arns Steiner, "Ten-year evolution on credit risk research: a Systematic Literature Review approach and discussion," Ingeniería e Investigación, vol. 40, no. 2, pp. 50-71, 2020.

[27] G. Teles, J. Rodrigues, R. A. Rabê, and S. A. Kozlov, "Artificial neural network and Bayesian network models for credit risk prediction," Journal of Artificial Intelligence and Systems, vol. 2, no. 1, pp. 118-132, 2020.

[28] S. Deng, R. Li, Y. Jin, and H. He, "CNN-based feature cross and classifier for loan default prediction," in 2020 International Conference on Image, Video Processing and Artificial Intelligence, 2020, vol. 11584: International Society for Optics and Photonics, p. 115841K.

[29] N. Sopitpongstorn, P. Silvapulle, J. Gao, and J.-P. Fenech, "Local logit regression for loan recovery rate," Journal of Banking & Finance, vol. 126, p. 106093, 2021.

[30] M. R. Pradhan, S. Akter, and A. Al Marouf, "Performance Evaluation of Traditional Classifiers on Prediction of Credit Recovery," in Advances in Electrical and Computer Technologies: Springer, 2020, pp. 541-551.

[31] A. Min, M. Scherer, A. Schischke, and R. Zagst, "Modeling Recovery Rates of Small-and Medium-Sized Entities in the US," Mathematics, vol. 8, no. 11, p. 1856, 2020.

[32] A. Al Imran and M. N. Amin, "Loan Charge-Off Prediction Including Model Explanation for Supporting Business Decisions," in International Conference on Intelligent Systems Design and Applications, 2020: Springer, pp. 1281-1291.

[33] L. Li and D. Huang, "A Loan risk assessment model with consumption features for online finance," in Journal of Physics: Conference Series, 2021, vol. 1848, no. 1: IOP Publishing, p. 012068.

[34] N. Siddique and H. Adeli, "Simulated annealing, its variants and engineering applications," International Journal on Artificial Intelligence Tools, vol. 25, no. 06, p. 1630001, 2016.

[35] S. Parusheva, "Social Media Banking Usage From Banks' Perspective," International Journal of E-Business Research (IJEBR), vol. 15, no. 1, pp. 38-54, 2019.

[36] V. D. W. Aryanto, Y. Wismantoro, and K. Widyatmoko, "Implementing Eco-Innovation by Utilizing the Internet to Enhance Firm's Marketing Performance: Study of Green Batik Small and Medium Enterprises in Indonesia," in Disruptive Technology: Concepts, Methodologies, Tools, and Applications: IGI Global, 2020, pp. 1290-1307.

[37] J. Liu, C. Wu, and Y. Li, "Improving Financial Distress Prediction Using Financial Network-Based Information and GA-Based Gradient Boosting Method," Computational Economics, vol. 53, no. 2, pp. 851-872, 2019.

[38] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical report-tr06, Erciyes university, engineering faculty, computer …, 2005.

[39] Y. Xu, P. Fan, and L. Yuan, "A simple and efficient artificial bee colony algorithm," Mathematical Problems in Engineering, vol. 2013, 2013.

[40] I. Podlubny, Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications. Elsevier, 1998.

Journal of Information Systems and Telecommunication
Vol.10, No.2, April-June 2022, 151-160

http://jist.acecr.org
ISSN 2322-1437 / ESSN:2345-2773

A New High-Capacity Audio Watermarking Based on Wavelet Transform using the Golden Ratio and TLBO Algorithm

Ali Zeidi Joudaki[1], Marjan Abdeyazdan[2*], Mohammad Mosleh[1], Mohammad Khayrandish[1]

[1]. Department of Computer Engineering, Dezful Branch, Islamic Azad University, Dezful, Iran
[2]. Department of Computer Engineering, Mahshahr Branch, Islamic Azad University, Mahshahr, Iran

Received: 26 Apr 2021/ Revised: 16 Dec 2021/ Accepted: 14 Jan 2022
DOI:

Abstract

Digital watermarking is one of the best solutions for copyright infringement, copying, data verification, and illegal distribution of digital media. Recently, the protection of digital audio signals has received much attention as one of the fascinating topics for researchers and scholars. In this paper, we presented a new high-capacity, clear, and robust audio signaling scheme based on the DWT conversion synergy and golden ratio advantages using the TLBO algorithm. We used the TLBO algorithm to determine the effective frame length and embedded range, and the golden ratio to determine the appropriate embedded locations for each frame. First, the main audio signal was broken down into several sub-bands using a DWT in a specific frequency range. Since the human auditory system is not sensitive to changes in high-frequency bands, to increase the clarity and capacity of these sub-bands to embed bits we used the watermark signal. Moreover, to increase the resistance to common attacks, we framed the high-frequency bandwidth and then used the average of the frames as a key value. Our main idea was to embed an 8-bit signal simultaneously in the host signal. Experimental results showed that the proposed method is free from significant noticeable distortion (SNR about 29.68dB) and increases the resistance to common signal processing attacks such as high pass filter, echo, resampling, MPEG (MP3), etc.

Keywords: Audio Watermarking; Discrete Wavelet Transforms (DWT); High Capacity; TLBO Optimization Algorithm.

1- Introduction

With the rapid and increasing growth of the Internet and the spread of digital multimedia technologies, it has become very easy to copy and exchange digital multimedia data. For this reason, data protection against copyright and digital multimedia data has become an important issue. One of the ways to prevent these problems is to use the watermarking technique. In watermarking, a hidden signal called a watermark is embedded directly inside the host signal and always stays inside it unrecognizably. This technology can be used in various fields such as audio, video, multimedia, etc. [1-2] and since the human auditory system is more sensitive than its visual system, it is more difficult to embed watermark bits clearly in the host signal (audio signal) than video and multimedia [3-4]. This hiding of the watermark signal should not reduce the quality of the host signal. The difference between watermarking and cryptography is that in watermarking, the information was used in the same way after the watermarking operation and it is not necessary to extract the watermark signal bits to use them. While in encryption, the encrypted signal is not usable and its use requires the extraction of the password. In general, watermarking operations do not restrict access to information. However, the purpose of cryptography is to restrict the access of unauthorized persons to information [5]. The important features of a reliable and efficient watermarking process are transparency, flexibility, and capacity. In a transparent watermark system, hiding the watermark signal does not affect the quality of the original signal. In the case of audio watermarking, the audio quality of the original audio signal and the watermarked audio signal should not be significantly different for the average listener. According to the International Federation of Sound Industry (IFPI), a signal should operate with a signal-to-noise ratio (SNR) greater than 20 dB [6-7]. Resistance: The watermarking system is resistant to common types of attacks and common processes. An audio watermarking system must be resistant to attacks such as MP3 compression, adding echoes, changing the sampling rate, and converting digital to analog and analog to digital.

✉ **Corresponding Author**
Abdeyazdan87@yahoo.com

Capacity: Refers to the number of bits that are embedded in a medium or in a unit of time in that medium. The storage volume of a photo is equal to the number of bits in which the photo is stored. But the capacity of an audio signal refers to the number of bits that audio is stored per second. In marking methods, a compromise must always be made between the above factors [8]. So far, no specific method has been proposed that can simultaneously meet all three requirements detailed above. Audio watermarking methods are always optimized in terms of the mentioned requirements. Moreover, after performing the watermarking operation, we have to extract the watermark signal [9]. Watermark signal extraction methods can be divided into the following three categories:

- Blind extraction: In this type of extraction, the main signal is not needed and the watermark mark is extracted directly from the watermarked signal. The algorithm proposed in this paper was falls under this extraction category.
- Semi-blind extraction: where some of the main signal properties are required for extraction.
- Non-blind extraction: This method requires the main signal for extraction [10-12].

Methods that use blind extraction are more applicable than non-blind methods. Fallahpour et al. [13] proposed an audio signal watermarking algorithm by embedding and extracting data by the FFT spectrum resizing method. The main point was to choose a frequency band for embedding based on the comparison between the compressed signal and the original and MP3 and a scaling factor. Their results showed the embedding of 5000 bps. Furthermore, the average signal to noise (SNR) was higher than 20 db. The ODG parameter is was also calculated in this paper. The value obtained for this parameter was -0.25. Prayoth Kumsawat [14] presented a blind, robust, transparent audio signal watermarking method with acceptable load capacity. They took advantage of artificial intelligence and wavelet transform. Watermark bits were embedded in the low-frequency coefficients of the signal determined by discrete wavelet transform. The embedding method was developed through the quantization process. A genetic algorithm was also used to optimize the quality of the watermarked signal. Their results confirmed the embedding of 34.14 bps. The average signal to noise (SNR) in this article was higher than 27.50 db. Meanwhile, the average BER value extracted in this study was about 0 to 4.0039%. The most important limitation of this system was its very low capacity. Martinez et al. [15] proposed a semi-blind, resistant, transparent audio signal watermarking method. They used the benefits of the LDPC technique. The watermark signal bits were embedded in the fifth level of the host audio signal specified by the wavelet transform. Their results demonstrated the embedding of 229 bps. The average signal to noise (SNR) in this article was higher than 40 db. Meanwhile, the average BER value was about 1.233×10^{-4}%. The most important limitation of this system was its very low capacity. Bath et al. [16] proposed a blind, transparent, and strong audio signal watermarking method.They used the advantages of SVD and QIM (quantization). The watermark insertion and extraction methods were designed based on quantization of the norm of singular values of the blocks. Their results showed the embedding of 196 bps. Besides, the average signal to noise (SNR) was higher than 20 db. Meanwhile, the average BER value was about 0 to 0.56&. The most important limitation of this system was its very low capacity. Fallahpour et al. [17] presented a blind extraction audio watermarking method using human auditory amplitude logarithmic domain (HAS). Their results showed the embedding of 7000 to 8000 bps. The average signal to noise (SNR) varied from 21 to 36 db. Meanwhile, the average BER value was about 0 to 13%. The ODG parameter was also calculated in this paper with the corresponding value of-1 to -0.1. Mohsenfar et al. [18] presented a semi-blind, robust, transparent sound signal watermarking scheme with adequate load capacity. They took advantage of QR decomposition (QR factorization) and genetic algorithms. They split the main audio signal into several frames. Each frame was then decomposed using the QR decomposition method, and then the most suitable location for embedding watermarks signal bits with high resistance to potential attacks was determined using a genetic algorithm. Their results confirmed the embedding of 159 bps. The average signal to noise (SNR) in this article was higher than 24 db. Meanwhile, the average BER value was about 0 to 24.18%. The ODG parameter was another calculated in this paper, with a value of -0.36 to -0.81. The most important limitation of this system was its very low capacity. Hu et al. [19] presented a blind, robust, transparent sound signal watermarking scheme with acceptable load capacity. They used variable dimensional vector modulation (VDVM) and the benefits of DWT to balance the two main parameters of capacity and transparency. Their results demonstrated the embedding of 301.46 bps. The average signal to noise (SNR) in this article was higher 20.400 db. Meanwhile, the average BER value extracted in this study was about 1.05%. ODG was also calculated in this paper. The value obtained for this parameter was -0.151[+-0.133]. The most important limitation of this system was its very low capacity. Chen et al. [20] proposed a semi-blind digital audio signal watermarking method. They embedded the watermark signal in the low-frequency coefficients of the discrete wavelet transform. They also used the Karush-Kuhn-Tucker (KKT) advantage to minimize the difference between the main signal coefficients and the watermarked signal. Their results showed the embedding of 1000 to 2000 bps. The average signal to noise (SNR) in this article was 20.21 db. Meanwhile, the average BER value was about 0.11%. Uma R. Nair et al. [21] presented a blind, robust, transparent sound signal watermarking scheme with a high capacity. They used the benefits of discrete wavelet transform (DWT).

First, they processed the original audio signal using wavelet transform. They then divided them into frames of appropriate size. Fibonacci numbers were used to embed the watermark signal bits. Their results confirmed the embedding of 2100 to 3125 bps. The average signal to noise (SNR) was higher than 58 to 69 db. Moreover, the average BER value extracted in this was about 0.0059%. Erfani et al. [22] proposed an audio signal watermarking method. The benefits of the Fourier domain were used to embed the watermark signal by changing the phase of the signal coefficients. Moreover, watermark bits are embedded in the upper domain. Their results showed the embedding of 56.5 bps. The average signal to noise (SNR) in this article was 20.1 db. Meanwhile, the average BER value was about -0.23 to -1%. Mustapha Hemis et al. [23] presented a sound signal watermarking method, semi-blind, robust, transparent, and with appropriate capacity. Using discrete wavelet converter (DWPT) and (DC-DM) converters, they were able to use DWPT to split audio frames into multiple frequency bands. Use the psychoacoustic model to determine the appropriate sub-bands for embedding watermark signal bits. Moreover, the DC-DM method was used to embed watermark bits in DWPT coefficients. For our compromise between strength and capacity, the sync code technique was used to resist synchronization attacks in the proposed method. Their results showed the embedding of 2500 bps. The average signal to noise (SNR) in this article was higher 35.95 db. Meanwhile, the average BER value was about 0 to 0.35%. The most important limitation of this system was its very low capacity. Arashdeep et al. [24] proposed a sound signal watermarking method with blind extraction, using the advantages of the wavelet transform. Their results showed the embedding of 4884 bps. The average signal to noise (SNR) in this article was 19.88 to 37.92 DB. Meanwhile, the average BER value was smaller than 0 to 5.117%. Sadie et al. [25] Proposed an audio signal marking scheme with blind extraction. Arnold's conversion was used to maintain detection security. They first segmented the host signal and then used the advantages of discrete wavelet transform (DWT) and discrete cosine conversion (DCT) per frame. Their results showed the embedding of 41.19 to 53.87 bps. The average signal to noise (SNR) in this article was 31.0786 db. In addition, the average BER value varied from 00. to 5.0781%. The most important limitation of this system was its very low capacity. Pourhashemi et al. [26] presented a new blind, resistant, transparent sound marking design with high capacity. Using the advantages of Lucas' regular mathematical sequence, they also proposed a 2-bit embedding method. An intelligent recursive adjustment process was also employed to determine the frame size and frequency band values. Their results approved the embedding of 1 to 8000 bps. The average signal to noise (SNR) in this paper varied from 33 to 58 db. Additionally, the average BER value was about 4.908%. The ODG parameter was estimated and the resulting value varied from -0.35 to -1.57. The main weakness of is the proposed design was its low resistance to some attacks. Mehdi Mosleh et al. [27] presented a new blind audio watermarking design that could use the synergy between fuzzy inference system, one-value analysis (SVD), and the Fibonacci sequence in discrete cosine conversion (DCT) between power transparency and compromise and create capacity. Their results demonstrated an embedding of 593.34 bps. The average signal-to-noise ratio (SNR) in this paper was 49.8093 db. Moreover, the average BER value was 1.3644. ODG was another parameter calculated in this paper with the resulting value varying from -0.18 to -0.46. The main disadvantage of the design developed in this article was its low capacity. Ahmad Hussein Ali et al. [28] proposed a high-capacity audio steganography model based on fractal coding and uniform coefficient modulation. Their model was able to use the HASFC approach based on latent coating mapping, uniform coefficient modulation, and hybrid turbulence mapping techniques in the field of wavelet transform. The average signal-to-noise ratio (SNR) in this paper was 50 db. Moreover, the average BER value was 0.0369. The ODG parameter calculated in this paper varied from 4.6 to 4.8. The main disadvantage of the developed model was its very low capacity as well as its low resistance to some attacks.

1-1- Discrete Wavelet Transformation (DWT)

The authors used the advantages of the TLBO algorithm to determine the frame length and the optimal location of the signal bits, and the optimal target performance in the embedding and extraction steps. In a marked signal, SNR was increased in the embedding process and the bit error was reduced in the extraction process. The main idea was to embed an 8-bit signal simultaneously in the host signal. As mentioned above, one of the most important challenges in the auditory signal system is to provide an effective way to increase the correlation between the three criteria of transparency, strength, and load capacity. To this end, this study presented a new audio signal scheme based on the benefits of DWT and the TLBO optimization algorithm, and the golden ratio. The other sections of the article are structured as follows: In the first section, the concept of watermarking and its applications are introduced. In the second section, the DWT conversion, the golden ratio, and then the TLBO algorithm are discussed. In the third section, the proposed method is introduced in detail. Simulation results and experimental data are presented in the fourth section of the paper. Finally, the article ends with a concluding section.

2- Preliminaries

The general outline of this paper is as follows: Initially, the host program reads the audio signal and the signal (image) separately. In the second step, wavelet transform is applied to the main audio signal in three levels and CD3 coefficients are generated. Then, using the TLBO algorithm, the CD3 coefficients are framed and the amount of embedding distance is also determined. Then, using the golden ratio sequence, the signal bits are embedded in appropriate suitable place in the host signal. Finally, a watermark signal is generated. In the extraction step, the specified signal (Lena image) is extracted. An inverted wavelet transform is applied to the specified signal in three steps. The coefficients below the high-frequency band are also extracted and framed. Finally, the binary equivalent of the extracted data is calculated and the pattern signal is extracted. signal at different scales.

Wavelet transform, like Fourier transform and cosine transform, models the data [29]. compute coefficients to indicate how similar a particular function is to applied data. Most wavelet analyses add a parameter that specifies how often this function is applied to the wavelet function, and the remaining intervals of the function are zeroed by this wavelet. What does not make sense in the sine and cosine waves is a powerful and flexible tool for solving many complex problems. Here are some of the features listed below: • Wavelet is a tool that can display a signal with different degrees of resolution and can be used to examine the signal at different scales. • Wavelet transform can display the signal in only a few non-zero sentences. The principles of discrete wavelet transforming go back to a method called sub-band bandwidth [30-31]. In the discrete mode, filters with different cut-off frequencies are used to analyze the as the signal passes through the upper and lower filters, its different frequencies are analyzed. In discrete wavelet transform, the signal resolution is controlled by the operation of the filters, and the scale changes with low sampling or high sampling. Processing begins with a discrete wavelet transform. Initially, the signal passes through a low-voltage digital half-band filter with a shock response of h [n], so the filter output is equal to the input convolution and the filter shock response. As a result of this filtering operation, all frequency components greater than half of the highest frequency in the signal are removed. Since the maximum output signal frequency of the filter is equal to π/2 radians, half of the samples can be omitted. Therefore, by deleting one of the samples, the signal length is halved without losing any piece of information. A similar process is performed using a semi-band digital high pass filter with a shock response of g [n]. As a result, at the output of the first stage of the wavelet conversion operation, two versions, one high-pass and one low-pass, with a reduced length (half) of the initial signal are obtained as follows:

$$y_{\mathrm{high}}[k] = \sum_n x[n].\, g[2k-n] \qquad (1)$$

$$y_{\mathrm{low}}[k] = \sum_n x[n].\, h[2k-n] \qquad (2)$$

This halves the temporal resolution and doubles the frequency resolution. This process can be re-applied to the low-pass version, doubling the frequency resolution at each stage by reducing the time resolution to half of the previous stage. This idea is known as the filter bank method for calculating discrete wavelet transform. The output coefficients of the low-pass filter follow the original shape of the signal. Hence, these coefficients are called approximations. High pass filter output coefficients also include high-frequency details. Hence, these coefficients are called details. As the number of conversion steps increases, so does the amount of details. It should be noted that the number of steps required to convert a discrete wavelet depends on the frequency characteristics of the analyzed signal. Finally, discrete wavelet wave conversion is obtained by juxtaposing the filter outputs from the first stage of the filter. Therefore, the number of wavelet conversion coefficients will be equal to the number of input discrete signal samples. Three levels of discrete wavelet transform decomposition are shown in Figure 3. As can be seen, A1 and D1 are the coefficients of the first level of signal decomposition X [n]. In the second level, A1 is decomposed into A2 and D2 and, in the third level, A2 is decomposed into A3 and D3:

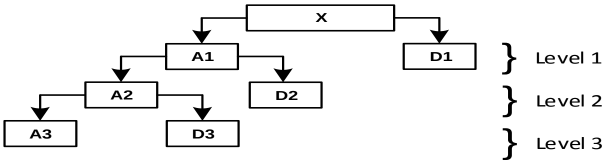


Fig.1 The three levels of the discrete wavelet transform decomposition

2-1- Golden Ratio

In mathematics and art, the golden ratio occurs when the ratio of the larger part to the smaller part is equal to the ratio of the whole to the larger part. It geometric definition was used to obtain the golden ratio, $\frac{a+b}{a} = \frac{a}{b} = \emptyset$. Following this equation, which is the definition of the number $(\emptyset)$, we can conclude from the equation on the right: $a = b \times \emptyset$. Thus, we will have: $\frac{b\times\emptyset+b}{b\times\emptyset} = \frac{b\times\emptyset}{b}$. By removing b from both sides., we have: $\frac{\emptyset+1}{\emptyset} = \emptyset$. The positive answer is as follows:

$$\emptyset = \frac{1+\sqrt{5}}{2} \simeq 1.6180339887 \qquad (3)$$

2-2- TLBO (Tteacher Learning Based Optimization) Algorithm

In recent years, meta-heuristic algorithms have been used to optimize engineering problems. These algorithms are either modeled based on natural phenomena (such as ant colony and birds algorithms) or sample human social exchanges (such as Imperial competition algorithms and teacher learning algorithms). The most important advantage of these algorithms is that they are simple and do not require complex mathematical problems such as derivatives and integrals. Teacher learning based optimization algorithm is an interesting algorithm for optimizing engineering issues which are modeled based on the teacher training in the classroom. This algorithm has two training steps. The first step is based on the teacher training and the second one is based on the student debate after the end of the class. In the first phase, the person who provides the best answer in the population is selected as the teacher ($X_{teacher}$) and other members of the population are known as students (X_i). In the following section, the average position of the students (X_{mean}) is calculated. The reason for calculating the student knowledge average is that the teacher gives the training according to the average level of the class. By considering “r” as a random number and T_f as a constant coefficient, it is possible to model the movement of students in the first step by the following equation [32-34].

$$\mathrm{X_{new}} = \mathrm{X_i} + \mathrm{r} \times (\mathrm{X_{teacher}} - \mathrm{T_f}.\mathrm{X_{mean}}) \tag{4}$$

Where X_i and X_{new} are the current and the new situation of the students, respectively, and Tf is a training factor that is considered as 2.

In the second stage, the teaching process is assigned to the students, so that each student selects another student randomly and shares knowledge with each other and also updates his/her position; thus trying to use the other students' information to raise his/her level of awareness and knowledge. This phase can be modeled using the following formulations [13-16]:

$$X_{new} = X_i + r(X_i - X_j) \quad if \quad f(X_i) < f(X_j) \tag{5}$$

$$X_{new} = X_i + r(X_j - X_i) \quad if \quad f(X_j) < f(X_i) \tag{6}$$

In this stage, the move is made if the new position is better than the previous position. Moreover, the condition for the termination of this algorithm is to reach the end of the iteration. The algorithm will continue until the termination condition is met. The method proposed in this study is a blind audio watermarking technique that is developed by applying the discrete wavelet transform (DWT) on the digital audio signal. The algorithm consists of two procedures; The watermarking embedding procedure and watermarking extraction procedure.

3- The Proposed Watermarking Scheme

The method presented in this article was a new approach to increasing the encryption capacity of audio signals for covert telecommunication purposes. In a wavelet transform using two filters, the signal is decomposed into several sub-bands in a specific frequency range. Because the human auditory system is not sensitive to changes in high-frequency bands, we use these sub-bands to embed blue signals. This increases the embedding capacity. The method discussed in this paper uses high-frequency coefficients above the surface of the third wavelet, to which the human ear is less sensitive than embedded blue markers. As mentioned, we need to find a way that fits the three criteria of transparency, flexibility, and capacity. Determining the appropriate sub-band plays a significant role in enhancing capacity and transparency. In this paper, we have determined the appropriate sub-band by wavelet transform. Moreover, the size of the frame is directly related to the amount of resistance. Therefore, the longer the frame, the more resistant it will be and the less capacity it will have, and vice versa. The appropriate frame length was determined using the TLBO algorithm. To further increase the resistance, the coefficients of this sub-band were framed and the average length of each frame was calculated and placed in the first cell of each frame. The signal was converted to a bit sequence, and each of the eight signal bits was placed in the appropriate range determined by the TLBO algorithm using the golden ratio sequence.

3-1- Watermark Embedding Procedure:

In this section, a sound-blind marking scheme with high-capacity and clear and robust features is developed based on the TLBO algorithm and wavelet transform. The embedding process is performed through 6 main operations:

Inputs: Watermark signal, suitable sub-band coefficients by DWT The frame length size is determined by the TLBO algorithm.

Output: Watermark signal the proposed method is detailed as follows:

Begin

Step 1: Before inserting the pattern into audio, the pattern is converted into a one-dimensional bit sequence. To convert bit patterns into a sequence of bits, we need to convert a two-dimensional array into a one-dimensional array. To do this, we scroll the bit pattern line by line and convert the bit values of the bit pattern into a bit sequence.

Step 2: We used the Dobchi 8 wavelet transform to analyze the host sound. Host sound analysis is performed at 3 levels. The lower the resolution level, the lower the resistance of the algorithm and the higher the watermark capacity, and vice versa, because the human ear is less sensitive to the coefficients of detail (high frequency). In this method, we model on the CD3 sub-band.

Step 3: The CD3 coefficients are divided by the TLBO optimization algorithm into the number of frames of a given length. Then, the average (mi) for each frame is calculated based on the average absolute values of the samples per frame. It should be noted that the greater the number of frames or the smaller the frame size, the greater the transparency of the algorithm and the lower the resistance, and the smaller the number of frames, the greater the power of the algorithm. Then, the average absolute value of the samples of each frame is calculated using the formula (7) and store it in the first cell of each frame.

$$\mathrm{m_i} = \frac{1}{\mathrm{s}} \sum_{\mathrm{j}=(\mathrm{i}-1)\mathrm{s}+1}^{\mathrm{i}\times\mathrm{s}} |\mathrm{C_j}| \qquad (7)$$

Where (C_j) is the sub-band detail coefficient of the third wavelet surface, s is the size of each frame, and (m_i) is the mean of the (i) frame. Using Equation (8) below, the number of watermark signal bits is calculated and placed in the third level coefficients. The parameter k is the embedding distance, which is less than one. If the value of this parameter increases, the amount of embedded capacity increases leading to a severe decrease in the power of the algorithm. In this paper, the advantages of the TLBO algorithm are used to select the optimal value of K factors that have reached the signal in proportion to the noise. The signal to noise ratio (SNR) is a measure of statistical difference used to determine the similarity between the original distorted audio signal and the distorted audio signal.

Step 4: Using the golden ratio, the appropriate locations of each frame for embedding watermark signal data are determined.

Step 5: The number of bits to be embedded is determined via Eq. (10). The power of the proposed algorithm is that at each appropriate factor (n) bits of the bit string of the watermark signal are embedded. This idea increases the cryptographic capacity. For example, in this paper, we consider the value of n = 8. The embedding capacity is equal to 2^n . The distance between $(K \times m_i)$ and $-(K \times m_i)$ is dividedby 256 levels, and then the decimal equivalent of each of the 8 watermark signals is stored in the variable eq. In the reconstruction process, the binary equivalent of the variable eq is used to reconstruct the desired printing mark.

$$\mathrm{C(i)} = -k \times \mathrm{m(i)} + \frac{2\times\mathrm{k}\times\mathrm{m(i)}\times\mathrm{double(xn)}}{2^{\mathrm{n}}} \qquad (8)$$

Step 6: Finally, after applying the embedding process using the reverse wavelet converter, a watermark signal is generated.

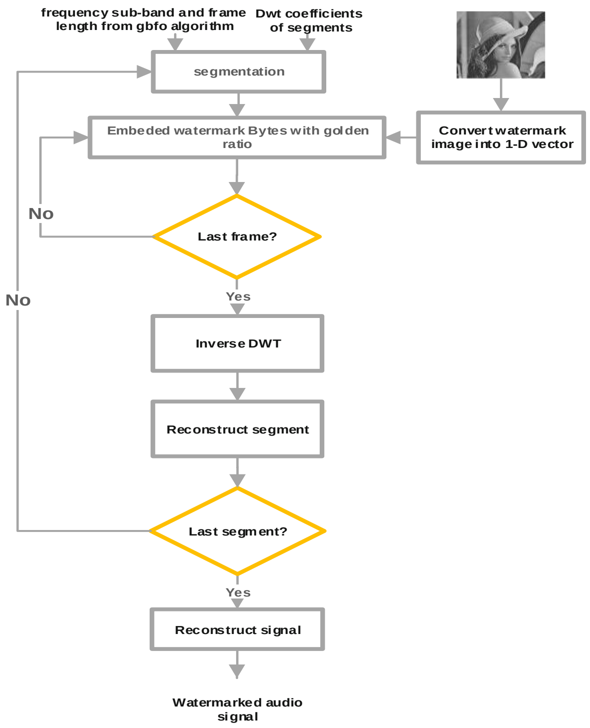

frequency sub-band and frame length from gbfo algorithm
Dwt coefficients of segments
segmentation
Embeded watermark Bytes with golden ratio
Convert watermark image into 1-D vector
No
Last frame?
Yes
No
Inverse DWT
Reconstruct segment
Last segment?
Yes
Reconstruct signal
Watermarked audio signal

Fig.2 The watermark embedding procedure

3-2- Watermark extraction procedure

The detection process is the inverse signal of the embedding operation, this process is blind because it does not require a host audio signal. Two main operations are performed at this stage: Frequency analysis based on discrete wavelet transform and cryptographic detection of the third level details of the audio signal. The detection process can be summarized in the following steps:

Inputs: The received audio signals[n]

Outputs: The extracted watermark

Begin

Step 1: The watermark audio signal received s [n]

Step 2: Three levels of wavelet transform are applied to the watermarked signal.

Step 3: After extracting the sub-band coefficients from the details of the third level, they are framed according to the fact that in the pasting stage, after reformatting and converting decimal data, the calculated average was stored at the location of the first cell of each frame. Decimal data stored in its binary equivalent can retrieve our data.

Step 4: Using the following equation (9), the string of cryptographic bits embedded in the signal is detected.

$$Ex = (C'(i\,,j) + K \times C'(i,1)) \times \frac{(2^8)}{2\times K\times C'(i\,,1)} + K \qquad (9)$$

C' High-frequency sub-band samples of the third level of watermark signal wavelet decomposition.

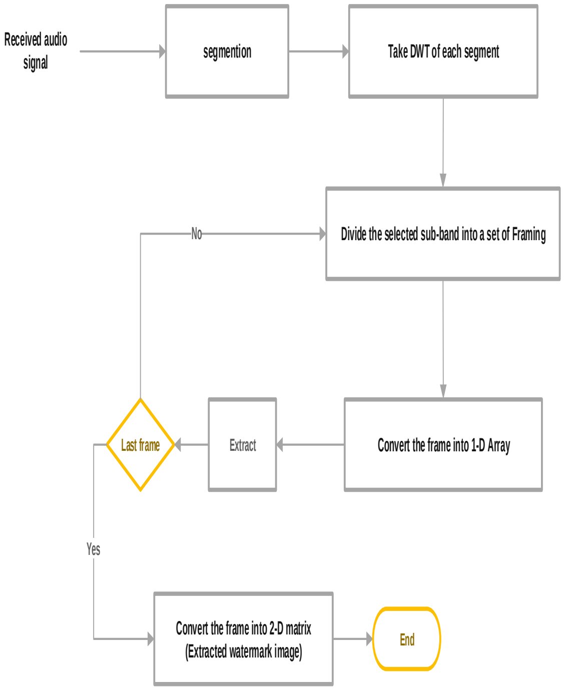

Received audio signal
segmention
Take DWT of each segment
Divide the selected sub-band into a set of Framing
No
Last frame
Extract
Convert the frame into 1-D Array
Yes
Convert the frame into 2-D matrix (Extracted watermark image)
End

Fig.3 Block diagram of watermark extraction algorithm

In this paper, we define the objective function as comparatively equal to Formula 10.
The teacher training optimization algorithm performs the optimization by minimizing the objective function. Thus, the objective function is defined as follows:

$$Fitness = A_1 \times SNR - A_2 \times BER + A_3 \times NC \qquad (10)$$

$$A_1 = 1 + \frac{\beta_1}{1+e^{\frac{SNR}{\alpha_1}}} \qquad (11)$$

$$A_2 = 1 + \frac{\beta_2}{1+e^{\frac{-BER}{\alpha_2}}} \qquad (12)$$

$$A_3 = 1 + \frac{\beta_3}{1+e^{\frac{NC}{\alpha_3}}} \qquad (13)$$

In the above equation, the coefficients A_1, A2, and A_3 which are related to the parameters SNR, BER, and NC are variable and they adapt their program function to the objective function. According to coefficient A_1, if the SNR value decreases for any reason, the denominator of the fraction decreases in the same way. As a result, A_1 increases, and the algorithm increases the SNR. In coefficient A_3, if the NC value decreases for any reason, the denominator of the fraction decreases in the same way. As a result, A_3 increases, and the algorithm increases the NC. But the coefficient A_2 works the opposite of the coefficients A_1 and A_3. Thus, if the value of BER increases, the denominator of the fraction also decreases and thus the coefficient A_2 increases by a negative sign, indicating that the algorithm decreases the value of BER. The values of coefficients $\beta_1, \beta_2, \beta_3$ are selected experimentally equal to 1, 50, and 20, and coefficients $\alpha, \alpha_2, \alpha_3$ are set as 25, 0.05, and 1, respectively.

4- Experimental Results

The method described in the previous section was implemented with Matlab 2019 software. This section presents the results of various tests performed on the proposed method. First, the sounds on which the cryptography was performed are identified [29]. These sounds are selected as common sounds on which other methods of cryptography are used in the experiments. We used the Lena image as the watermark image shown in Figure 4:

Fig.4 The watermark image

Loopy Music is an audio signal in which the carrier signal is chosen. To implement the watermarking procedure, an audio signal with 44100 HZ frequency is sampled and quantized with 16 bits. Then, by applying a three-level wavelet transform, the audio signal is divided into four sub-bands called are CA3, CD3, CD2, and CD1, respectively. Here, the CD3 sub-band was used to hide watermarks. We used the 8-coefficient Daubechies wavelet (db8) to decompose the audio signal. The simulation results also showed that by the best robustness against attacks would be achieved using Daubechies filter (db8). Using Equation 14, the signal-to-noise (SNR) value can be calculated. As it is clear from the formula, this parameter shows the similarity of the watermarked signal to the main signal. According to the International Federation Phonographic Industry (IFPI) standard, the SNR value must be at least 20dB and is calculated using Equation 14:

$$SNR = 10 \log_{10} \frac{\sum_{n=0}^{N-1} X_O{}^2(n)}{\sum_{n=0}^{N-1}[X_O(n) - X_w(n)]^2} \qquad (14)$$

In Equation (14) $X_O(n)$, $X_w(n)$ signals are the main signal and the watermarked signal, respectively. The minimum acceptable value for this parameter is 20 db.

4-1- Subjective Testing

We must test the final quality of the results of the original signal and the watermarked signal by the human auditory system. To do this, we select five volunteers and ask them to tell us the difference between the original signal and the watermarked signal. Degree of mental difference Degree (MDG). In Table 1, MDG = 5.0 has the concept of excellent quality and degree MDG = 1.0 indicates poor quality.

Table 1 ODG and MDG values with their descriptions

explanation	Quality	MDG values	ODG values
Inconspicuous	Excellent	5.0	0.0
Tangible	Good	4.0	-1
Some Pesky	Exhibition	3.0	-2
Pesky	Impaired	2.0	-3
Vary Pesky	Bad	1.0	-4

4-2- Objective Testing

The degree of objective difference (ODG) and the signal-to-noise ratio (SNR) are two important parameters for the objective evaluation of watermarked signals. In this article, we intend to compare and evaluate the proposed method with a focus on the previously presented methods and an assessment of these parameters. Using Equation 14, the signal-to-noise (SNR) ratio can be calculated. As is shown, this parameter shows the similarity of the watermarked signal to the main signal. According to the International Federation Phonographic Industry (IFPI) standard, the SNR value must be at least 20 dB according to Equation 14. The SNR values of the different signals that have been watermarked by the method presented as can be seen in Table 3. The ODG parameter is measured with EQUAL software [30]. The value of this parameter can vary from -4 (indicating that the applied changes are very annoying) to 0 (indicating that the applied changes are imperceptible). The SNR values of the different signals that have been watermarked by the method presented as can be seen in Table 2:

Table 2 The results of objective and subjective tests performed on the proposed method

Audio files	Objective tests	ODG	Subjective test
	SNR (dB)		MDG
Blues	29.42	- 0.56	4.19
Electronics	30.31	- 0.16	4.25
Average	29.32	-0.321	4.36

4-3- Robustness Testing

The BER (bit error rate) and NC (normalized correlation) are two important parameters used for evaluating the resistance of watermarked signals. Equations 15 and 16 show how to calculate these two parameters:

$$NC = \frac{\sum_{i=1}^{M_{lenght}} \sum_{j=1}^{M_{width}} [M(i,j)M'(i,j)]}{\sum_{i=1}^{M_{lenght}} \sum_{j=1}^{M_{width}} [M(i,j)]^2} \qquad (15)$$

$$BER\ (W\ ,\bar{W}) = \frac{\sum_{i=1}^{M} \sum_{j=1}^{M} W(i,j) \oplus \bar{W}(i,j)}{M \times M} \qquad (16)$$

Table 3 The assessment of the resistance of the proposed design against Stir-mark attacks for BER and NC

Audio file	Attack Type	NC	BER (%)
Electronic	Nothing	0.9989	0.1879
	Lsbzero	0.9831	0.9412
	fft_real_reverse	0.9642	0.6721
	fft_invert	0.9721	5.3446
	Addsinus	0.9916	0.9412
	addbrumm_2100	0.9933	0.7514
	rc_highpass	0.9845	0.9654
Average		**0.9839**	**1.4005**
Blues	Nothing	0.9899	0.7325
	Lsbzero	0.9731	1.8123
	fft_real_reverse	0.9754	2.9478
	fft_invert	0.9768	1.0841
	Addsinus	0.9842	1.7367
	addbrumm_2100	0.9657	0.8579
	rc_highpass	0.9731	1.1124
Average		**0.9768**	**1.4689**

4-4- Data Ccapacity Result

Capacity refers to the number of bits stored per second of the host sound and is expressed as bits per second. Capacity is calculated via the following equation:

$$\text{payload} = \frac{\text{Total bits of watermark}}{\text{time of host audio}} \qquad (17)$$

Table 4 Comparison the proposed design with other methods in terms of capacity and intangibility

Watermarking schemes	capacity (bps)	Average of ODG	Average of SNR (dB)
[13]	8005	N/A	21.03
[14]	34.14	N/A	34.14
[15]	229	N/A	> 40
[16]	196	N/A	> 20
[17]	7000 to 8000	N/A	21 to 36
[18]	159	-0.36 to -0.81	24
[19]	301.46	-0.151[±0.133]	20.14
[20]	1000 to 2000	N/A	20.21
[21]	2100 to 3125	N/A	58 to 69
[22]	56.5	-0.21 to -1	> 20
[23]	2500	-0.35	35.95
[24]	4884	N/A	37.92
[25]	41.19 to 53.87	N/A	31.0786
[26]	1 to 8000	-0.35 to -1.57	33 to 58
[27]	598.34	-0.18 to -0.46	49.8093
[28]	562.4	4.6 to 4.8	50.4
Proposed	13000	- 0.56 to– 0.16	29.68
N/A means no report is found in the watermarking scheme			

Table 5 Comparing the proposed watermarking scheme with the previous works

Scheme	BER (%)	Payload capacity (bps)	Average of SNR (dB)	Blind	Average of ODG	Opti-method
[13]	0.0 to30	8005	21.03	Blind	N/A	N/A
[14]	0 to 4.0039	34.14	34.14	Blind	N/A	GA
[15]	10323×10^{-4}	229	> 40	Sami-Blind	N/A	N/A
[16]	0.0 to 0.56	196	> 20	Blind	N/A	SVD-Quantization
[17]	0 to 13	7000 to 8000	21 to 36	Blind	N/A	N/A
[18]	0 to 24.18	159	24	Sami-Blind	-0.36 to -0.81	GA
[19]	1005	301.46	20.14	Blind	-0.151[± 0.133]	VDVM
[20]	0.11	1000 to 2000	20.21		N/A	N/A
[21]	0.0059	2100 to 3125	58 to 69	Blind	N/A	N/A
[22]	0.0 to 5	56.5	> 20	Blind	-0.21 to -1	N/A
[23]	0.0 to 0.35	2500	35.95	Sami-Blind	-0.35	DWPT
[24]	0.0 to 5.17	4884	37.92	Blind	N/A	N/A
[25]	0.0 to 5.0781	41.19 to 53.87	31.0786	Blind	N/A	N/A
[26]	-0.32 to -1.57	1to8000	33 to 58	Blind	-0.35 to -1.57	N/A
[27]	1.3644	598.34	49.8093	Blind	-0.18 to -0.46	Fuzzy
[28]	0.0369	562.4	50.4	None-Blind	N/A	N/A
Proposed	1.4698	13000	29.68	Blind	- 0.56 to–0.16	TLBO
N/A means no report is found in the watermarking scheme						

5- Conclusion

This paper presented a new, high-capacity, DWT-based watermark audio signal-blind approach. In the proposed algorithm, to further increase power and transparency, the third-level sub-band coefficients were framed and then the average of each frame was calculated. The TLBO algorithm was used to determine the appropriate frame length and appropriate embedding range. In addition, the proposed method is blind method, which means that when the print signal is retrieved, the original signal is not required. The experimental results showed that the proposed method has an excellent SNR signal quality of about 29.68 dB and a very good capacity (about 13 kbps). It also has good resistance against sound processing attacks such as MP3 compression attacks, filtering, sampling reduction, etc.

References

[1] Swanson MD, Zhu B, Tewfik AH, Boney L. Robust audio watermarking using perceptual masking. Signal processing. 1998, pp.337-355.

[2] Liu W, Hu AQ. A sub-band excitation substitute based scheme for narrowband speech watermarking. Frontiers of Information Technology & Electronic Engineering. 2017, pp.627-643.

[3] Xia Z, Wang X, Zhang L, Qin Z, Sun X, Ren K. A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing. IEEE transactions on information forensics and security. 2016, pp.2594-2608.

[4] Nguyen TS, Chang CC, Yang XQ. A reversible image authentication scheme based on fragile watermarking in the discrete wavelet transform domain. AEU-International Journal of Electronics and Communications. 2016, pp.1055-1061.

[5] Sun L, Xu J, Liu S, Zhang S, Li Y, Shen CA. A robust image watermarking scheme using Arnold transform and BP neural network. Neural Computing and Applications. 2018, pp.2425-2440.

[6] slam M, Roy A, Laskar RH. SVM-based robust image watermarking technique in LWT domain using different sub-bands. Neural Computing and Applications. 2020, pp.1379-1403.

[7] Katzenbeisser S, Petitcolas FA. Digital watermarking. Artech House, London. 2000; 2.

[8] Nejad MY, Mosleh M, Heikalabad SR. LSB-based quantum audio watermarking using MSB as arbiter. International Journal of Theoretical Physics. 2019, pp.3828-3851.

[9] Rasti P, Samiei S, Agoyi M, Escalera S, Anbarjafari G. Robust non-blind color video watermarking using QR decomposition and entropy analysis. Journal of Visual Communication and Image Representation. 2016, pp.838-847.

[10] Bhardwaj A, Verma VS, Jha RK. Robust video watermarking using significant frame selection based on the coefficient difference of lifting wavelet transform. Multimedia Tools and Applications. 2018, pp.19659-19678.

[11] Ali M, Ahn CW. An optimal image watermarking approach through cuckoo search algorithm in the wavelet domain. International Journal of System Assurance Engineering and Management. 2018, pp.602-611.

[12] Mehta R, Rajpal N, Vishwakarma VP. Robust image watermarking scheme in lifting wavelet domain using GA-LSVR hybridization. International Journal of Machine Learning and Cybernetics. 2018, pp.145-161.

[13] Fallahpour M, Megias D. Robust high-capacity audio watermarking based on FFT amplitude modification. IEICE TRANSACTIONS on Information and Systems. 2010, pp.87-93.

[14] Kumsawat P. An efficient digital audio watermarking scheme based on genetic algorithm. In2010 10th International Symposium on Communications and Information Technologies 2010, pp. 481-485.

[15] Martínez-Noriega R, Nakano M, Kurkoski B, Yamaguchi K. High payload audio watermarking: toward channel characterization of MP3 compression. Journal of Information Hiding and Multimedia Signal Processing. 2011, pp. 91-107.

[16] Bath V, Sengupta I, Das A. A new audio watermarking scheme based on singular value decomposition and quantization. Circuits, Systems, and Signal Processing. 2011, pp.915-927.

[17] Fallahpour M, Megías D. High capacity logarithmic audio watermarking based on the human auditory system. In2012 IEEE International Symposium on Multimedia 2012, pp. 28-31.

[18] Mohsenfar SM, Mosleh M, Barati A. Audio watermarking method using QR decomposition and genetic algorithm. Multimedia Tools and Applications. 2015, pp.759-779.

[19] Hu HT, Hsu LY, Chou HH. Perceptual-based DWPT-DCT framework for selective blind audio watermarking. Signal processing. 2014, pp.316-327.

[20] Chen ST, Hsu CY, Huang HN. Wavelet-domain audio watermarking using optimal modification on low-frequency amplitude. IET signal processing. 2015, pp.166-176.

[21] Nair UR, Birajdar GK. Audio watermarking in wavelet domain using Fibonacci numbers. In2016 International Conference on Signal and Information Processing (IConSIP), 2016, pp. 1-5.

[22] Erfani Y, Pichevar R, Rouat J. Audio watermarking using spikegram and a two-dictionary approach. IEEE transactions on information forensics and security. 2016, pp.840-852.

[23] Hemis M, Boudraa B, Megías D, Merazi-Meksen T. Adjustable audio watermarking algorithm based on DWPT and psychoacoustic modeling. Multimedia Tools and Applications. 2018, pp.11693-11725.

[24] Kaur A, Dutta MK. An optimized high payload audio watermarking algorithm based on LU-factorization. Multimedia Systems. 2018, pp.341-353.

[25] Saadi S, Merrad A, Benziane A. Novel secured scheme for blind audio/speech norm-space watermarking by Arnold algorithm. Signal Processing. 2019, pp.74-86.

[26] Pourhashemi SM, Mosleh M, Erfani Y. Audio watermarking based on the synergy between Lucas regular sequence and Fast Fourier Transform. Multimedia Tools and Applications. 2019, pp.22883-22908.

[27] Mahdi Mosleh, Saeed Setayeshi , Behrang Barekatain, Mohammad Mosleh. A novel audio watermarking scheme based on fuzzy inference system in DCT domain. Multimedia Tools and Applications. 2021, pp. 20423–20447

[28] Ahmed Hussain Ali, Loay Edwar George, Mohd Rosmadi Mokhtar. An Adaptive High Capacity Model for Secure Audio Communication Based on Fractal Coding and Uniform Coefficient Modulation. Circuits, Systems, and Signal Processing. 2020. pp. pages5198–5225.

[29] Cai, Y.M., Guo, W.Q. and Ding, H.Y., 2013. Dwt-svd. Journal of Software, 8(7), p.1801.

[30] Brannock, E., Weeks, M. and Harrison, R., 2008, April. Watermarking with wavelets: simplicity leads to robustness. In IEEE SoutheastCon 2008, pp. 587-592.

[31] Pandey, V., 2014. Analysis of image compression using wavelets. International Journal of Computer Applications, 103(17).

[32] Lerch, A. (2002). Zplane development, EAQUAL-Evaluate Audio QUALity, version: 0.1. 3alpha. Retrieved July 2018, from http://www.mp3-tech.org/programmer/misc.html.

[33] Akhlaghi M, Emami F, Nozhat N. Binary TLBO algorithm assisted for designing plasmonic nano bi-pyramids-based absorption coefficient. Journal of Modern Optics. 2014, pp.1092-1096.

[34] Akhlaghi M, Emami F, Nozhat N, "Binary TLBO algorithm assisted for designing plasmonic nano bi-pyramids-based absorption coefficient", Journal of Modern Optics, 2014, pp.1092-1096.

[35] Akhlaghi M. Optimization of the plasmonic nano-rods-based absorption coefficient using TLBO algorithm. Optik. 2015, pp.5033-5037.

[36] Balvasi M, Akhlaghi M, Shahmirzaee H, "Binary TLBO algorithm assisted to investigate the supper scattering plasmonic nanotubes", Superlattices and Microstructures. 2016, pp.26-33.

[37] Kaboli M, Akhlaghi M, “Binary teaching-learning-based optimization algorithm was used to investigate the superscattering plasmonic nanodisk", Optics and Spectroscopy. 2016, pp.958-63.