International Academic Institute for Science and Technology

International Academic Journal of Science and Engineering
Vol. 12, No. 3, 2025, pp. 109-124.

ISSN 2454-3896

International Academic Journal of Science and Engineering

www.iaiest.com

An Optimized Feature Selection and Deep Learning Framework for Kawasaki Disease Detection in Clinical Data Mining

C. Kavitha [a], Dr.A. Subramani [b]

[a] *Research Scholar, Mother Teresa Women's University, Kodaikanal, Dindigul, Tamil Nadu, India. E-mail: kavithaphd2021@gmail.com, ORCID: https://orcid.org/0009-0008-8783-3676*
[b] *Assistant Professor, Department of Computer Science, M.V. Muthiah Govt. Arts College for Women, Dindigul, Tamil Nadu, India. E-mail: subramani.appavu@gmail.com, ORCID: https://orcid.org/0000-0002-8303-8770*

Abstract

Kawasaki disease (KD) is a form of acute vasculitis, which has been accompanied with coronary aneurysm of the artery, coronary artery dilation, arrhythmia and other deadly heart diseases. Even nowadays, its etiology is not clear enough; the molecular mechanism and the associated issues of KD should be investigated. Over the past few years deep learning has replaced the state-of-the-art in a number of tasks in medicine. The KNN (K-Nearest Neighbour) algorithm was the first algorithm that was used in imputing missing data. The optimization algorithm the Adapted Bee Colony with Firefly (ABCFF) is then used to do the feature selection i.e. selection of the best feature out of training data. In addition, the Firefly algorithm generates a new location where the new location of the scout bee stage will be replaced. Its functionality is tested against some benchmark functions. The f-score values that they consider as the best features selection criteria are the values of fitness applied in the present work. The parameter-optimized convolutional neural network (CNN) is used in the classification, and particle swarm optimization is used to figure out the best parameters of the CNN. The existing method can improve clinical data mining. The paper reported that a feature representation using matrices and feature extraction using CNN can be applied on mine clinical data which is left unfinished to facilitate the medical decision making.

Keywords: Kawasaki Disease, Deep Learning, K-Nearest Neighbours, Adapted Bee Colony with Firefly, Convolutional Neural Network, and Particle Swarm Optimization.

DOI: 10.71086/IAJSE/V12I3/IAJSE1253
Received: 30 June 2025/Revised: 07 August 2025/Accepted: 04 September 2025/Published: 30 September 2025

Introduction

Kawasaki disease (KD) (Darby & Jackson, 2021) was initially described in 1967 by a young pediatrician, Tomisaku Kawasaki, at the Red Cross Hospital in Tokyo (Burns & Glodé, 2004). Around the same time, Marian Melish described a similar clinical picture in a group of children in Hawaii, United States. KD is now reported worldwide, with some centers in India (Singh & Kawasaki, 2016). Given the epidemiology and clinical features, an infectious etiology has long been suspected, but no definitive causative agent has been identified to date. As with most other vasculitides, this diagnosis is made based on a temporal pattern of clinical manifestations, none of which is pathogenic in itself. It is thought that KD is the most prevalent childhood vasculitic disease. Well into the 60-150 per 100,000 children under the age of five years range, incidence rates have been reported in several countries. However, in India (as well as possibly in most other developing countries), most affected children with KD remain undiagnosed, likely due to ignorance among the pediatricians (Singh & Kawasaki, 2016).

The current diagnosis of KD is clinical and is backed by noninvasive laboratory examination. KD diagnosis is grounded in clinical guidelines in clinical practice that come up with epidemiological criteria, including a minimum of 4 of 5 major clinical findings and fever during 4 to 5 days. By doing so, KD becomes difficult to assess at the early stage. Besides, not all of the patients with incomplete KD correspond to the requirements and may even display some strange symptoms. The clinical features of KD resemble those of other febrile diseases and this is the reason why the percentage of missed KD diagnosis is high. Consequently, vulnerable and distinct methods that can be used to detect and treat early are also required to be investigated. In order to surmount the above issues and solve the problem of clinical decision-making, a data-driven approach to the appropriate and timely assessment of KD based on regularly reviewed, structured electronic health record (EHR) data was created, as well as a novel strategy of diagnosis.

EHRs give valuable and detailed information on individual health status, response to treatment, and clinical outcomes (Newburger et al., 2004). The existence of HERs in the prediction and diagnosis of clinical events is promising. Perotte et al. used periodically collected EHR data including longitudinal laboratory tests and clinical documentation to set up precise risk predictor models of the chronic kidney disease progression (Perotte et al., 2015). EHRs are used by Teoh et al. as a marker of a patient having a stroke during the 1-year time since the observation results and medical diagnosis (Teoh, 2018). Jin et al. used the one-hot coding and word representation to simulate the diagnosis incident in EHRs and forecast heart failure using a long short-term memory network (Jin et al., 2018). One of the potential methods of overcoming the weaknesses of a randomized clinical trial, such as small sample sizes, confounding factors, and comorbidities that cannot be subdivided, is the use of EHRs as clinical novelties (Yadav et al., 2018). EHRs can be used by intelligent systems to find new clinical evidence, implement knowledge in clinical practice, and provide better delivery of individualized healthcare.

EHRs are linked to numerous issues, and completeness, accuracy, complexity, and bias are the problems that can be attributed to the system despite its vast potential. The issue of data incompleteness is also widespread in the case of secondary analysis of EHR data and, therefore, limits the validity and relevance of the results. The overall approach of addressing the issue, which comprises the missing data, is data imputation. Substitution of the lost values or even rejection of the observations may be deceptive. A significant part of the previous research of data imputation has presupposed a small percentage of non-observation (e.g. less than 30 percent). The missing mechanisms are of three elementary types which are missing completely at random, missing at random and missing not at random. This is not always the case in clinical data that cannot be complete due to various factors and are described by a high degree of missing. The patient data can be exchanged among the facilities; therefore, the data in EHRs in one facility may be unfinished (Lin & Tsai, 2020). In certain medical scenarios, the physicians will be inclined to provide a patient with a generic list of health tests and laboratory tests. It might be medical tests or lab results that are not mandatory or that the information might be derived by knowledge of the medical personnel.

A number of researches on deep learning algorithms based on computer-aided diagnosis have shown successful outcomes. Cancer/lesion diagnosis systems based on deep learning have already been proven to be effective and in some fields, they have already surpassed human physicians (Esteva et al., 2017). Additionally, deep learning algorithms have been demonstrated to be more successful in the medical vision framework as compared to conventional ones (Liu et al., 2019). The deficiency of information in this work can also serve as a good source of information that can be utilized to determine the state of health of patients. The study also had the intention of establishing procedures to augment the implementation of clinical data mining in the incomplete-data environment, in addition to a missing-data imputation, to describe and integrate multiple sets of features. The motivation is summed up into two dimensions. To begin with, the incompleteness of clinical data, which is related to various clinical assessment measures, could be explained through feature selection and the presentation of a matrix. Second, PSO with

CNN has the ability to draw features out of unfinished information and even group a cluster of features to carry out beneficial classification.

Section 2 describes in brief current approaches to predicting KD; Section 3 gives the proposed approach to be used. Section 4 gives the comparison of the accuracy of the suggested method. Lastly, there is the conclusion and future work section 5.

Related Work

In the article by Luo & Li, (2022), they consider the diagnostic value of a deep learning algorithm in the coronary arteries of children with Kawasaki disease since a convolutional neural network (CNN) is used in analyzing the sources of the data of a dual-source computed tomography (DSCT) image of the coronary artery lesions in children with Kawasaki disease. The negative predictive value was 97.3 with positive predictive value of 92.7 which is more than the control. RISEU-Net-N, RISEU-Net and RIU-Net had a score of 0.7765, 0.8562 and 0.8356 respectively. It seems that this correlation between the results of the Optical coherence tomography (OCT) and the vessel distensibility of a new, simple deep learning, coronary artery segmentation system was confirmed by Benovoy et al., (2022) that automatically measured the change in coronary stiffness over time during a few years. However, there is no certainty that these results are associated with the change in the distensibility of coronary artery (CA) or the development of these changes.

The data introduced by Li et al., (2019) are informative on the phosphosignaling, and this information can be used to comprehend the molecular processes and possible targets of a new KD therapy. The phosphorylated proteins are studied to explain the applicable pathophysiological processes of the cardiovascular disease. However, phosphorylated proteins in patients with KD were studied not extensively. Given that the flow-mediated dilation (FMD) of the brachial artery declines in children with the development of KD, Wen et al., (2021) confirmed that it can be an early predictor of disease pathogenesis of CAA in children at the acute phase. They investigated its association with changes in the diameter of coronary arteries. It is not however obvious that endothelial dysfunction is associated with the extent of CAAs during the acute stage of KD.

Na et al., (2019) measured the size of left main coronary artery (LMA) and right coronary artery (RCA) against that of aorta (Ao), left anterior descending coronary artery (LAD), and aorta (Ao), respectively and determined the LMA /AO, LAD/ AO and RCA /AO ratios. The LMA/Ao cut-off level was 0.2, and the sensitivity of the test was 85 percent, but the specificity is 70 percent. The ratios of coronary artery/Ao in individuals might be useful in identifying coronary abnormalities in KD in the acute period. However, echocardiographic evidence of the dilation of the coronary arteries in KD is intermittent and does not invariably occur in all defective arteries. The gaps in group-oriented clinical data of different clinical assessment measures can be attributed to the lack of complete clinical data.

Ae et al., (2023) evaluated the variances and differences between the variables of the hospital background and the variables of the patient and compared the prevalence of CA abnormalities in groups of patients as per the criteria of Z-score. It was found that the Z score criteria was very sensitive in indicating patients with CA dilatations and no age and CA aneurysms in patients alone only in patients with age of 1 year and less as compared to the Japanese criteria. The article by Inagaki et al., (2019) constituted a retrospective study of the Kids Inpatient Database, which was conducted in 2006, 2009, 2012 and 2016. The codes of the International Classification of Diseases were employed to establish cases of Kawasaki disease and deep neck space in the patients under the age of 12 years. The outcome and demographic data were compared between the cases of deep neck space involvement of Kawasaki disease. Deep neck space connotation is connoted to be of increased cost and prolonged hospitalization and alleged abscess surgery.

The population of 10367 patients collected by means of electronic health records served as the population subject to the data-driven methodology developed in the research conducted by Wang et al., (2020) in estimating early KD. In addition, worked out an approach that involves feature clustering in order to enable matrices and convolutional neural networks (CNNs) to retrieve and add features to provide a certain benefit to the multi-source nature of the disease due to its unknown pathogenesis and absence of pathogenic features. Lee et al., (2022) aimed at testing the performance of a deep learning algorithm on the classification of KD and other acute febrile diseases. Methods: The echocardiographic images of water coronary arteries in children with KD (n = 138) and pneumonia (n = 65) have been received. The incidence of the lesions of the coronary arteries was high using the six deep learning networks (VGG19, Xception, ResNet50, ResNext50, SE-ResNext50, and SE-ResNext50), which were trained on the received data. Coronary artery lesions can be detected by echocardiography and this is what is relevant in the diagnosis of KD, as well as its positive prognosis.

Inference

A number of deep learning algorithms have been designed to detect a number of diseases such as breast cancer, liver cancer and thyroid nodules. It is implied that such medical deep learning algorithms enhance the work of the diagnostics via the computer. Deep learning algorithms on KD diagnosis have not been put into practice to such an extent. This paper set out to find out how the explicable deep learning algorithms can be utilized to detect lesions in the KD dataset in the view of providing an accurate diagnosis of KD with timely diagnosis (Park et al., 2016).

Proposed Methodology

The main aim of the research model is to create a smart system that will be able to conduct clinical data mining to detect KD more precisely as demonstrated in fig. 1. To begin with, the random forest method concerning the imputation of missing data was offered. KNN possesses everything that is required to address the problem of data imputation. The ABCFF technique is applied to pick the most optimal features in training datasets to select them. The best features are determined using f-score value. Through the proposed approach of hybridization, CNN-PSO, categorization is accomplished through clustering as a requirement and successive runs.

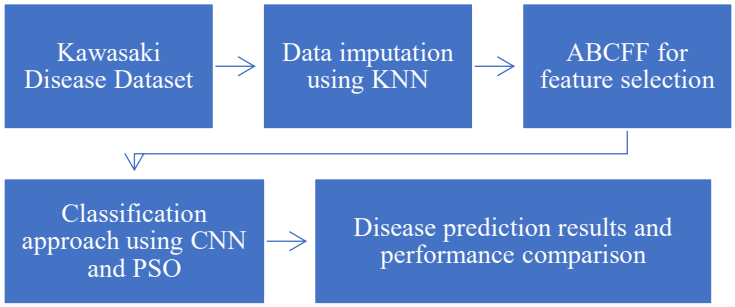

Fig. 1: KD Detection Process Using a Hybrid CNN-PSO

Collection of Data

The data set is 2752 cases of patient records that contain 15 columns describing the different aspects of Kawasaki Disease cases. It provides the epidemiological trends, treatment and post treatment outcomes in a broad perspective. Therapy methods such as IVIG and aspirin are also recorded and offer an insight on the common methods of treatment. The data on clinical outcomes after the treatment will give the indications of the effectiveness of the interventions, and the data about complications will help in the assessment of risks. The information about the follow-up visits assists in separating the post-treatment follow-up processes, whereas the data on the long-term effects will be used to learn more about the long-term effects of the disease. The dataset can be regarded as a useful resource to medical professionals and scientists working on the Kawasaki Disease and its treatment. Accessible in: https://www.kaggle.com/datasets/cm037divya/kawasaki-dataset.

Data Pre-Processing Using KNN for the Data Imputation Process

Pre-processing of data means dealing with missing points and then subjecting the data to machine learning algorithms such as K-Nearest Neighbours (KNN). Imputation KNN-based Imputation uses the values of neighbouring data points to fill in the missing values. The K-NN imputation algorithm is employed in the present research on a set of cases with different mechanisms and missing data models. Naive Bayes algorithm classification on this imputed data. K-NN method works as follows (Pujianto et al., 2019):

- **Determine The Parameter k**. In this study, k = 1, 3, 5, and 7 were used. There is no specific method for determining the k value for the K-NN method. If k is too small, there will be excessive noise, reducing classification accuracy. Still, if it is too large, it can also cause errors in limiting the value range and, by extension, affect accuracy (Pujianto et al., 2019).
- **Calculate The Manhattan Distance** between missing-data instances and complete-data instances using the equation.

$$d(x,y) = \sum_{j=1}^{s} x_j + y_j \qquad (1)$$

- Where $d(x,y)$ is the Manhattan distance, j: data attribute with j=1,2, 3,... s; s: data dimensions:x_j value from j-attribute containing missing data, y_j value from other j- attribute containing complete data.

- **Based On the Distance Information Obtained, The Minimum** Manhattan distance is calculated using the parameter k, which is set to the estimated value for the missing data. The value imputation is calculated using Weight Mean Estimation in equation (2)

$$x_j = \sum_{k=1}^{K} \frac{w_k v_k}{w_k} \quad (2)$$

Where x_j: Weight Mean Estimation, K: parameter number k selected, k=1,2, 3,....K, w_k: nearest neighbor observation value K; v_k: value in complete information on attribute with missing information depending on parameter k. In this case the w_k formula is computed in equation (3).

$$w_k = \frac{1}{d(x,y)^2} \quad (3)$$

To estimate the value of good or bad items using the imputation method, this study analyzes the accuracy obtained by various classification methods.

Feature Selection Using ABCFF Methods

The algorithm of ABC is computed by the steps of the ABC, and the result of the algorithm is called ABCFF. Search space division (SSD) is used to create the population (Panniem & Puphasuk, 2018). During the employed bee phase, ABC search equation is maximized with added more information on the best solution in a manner that the search is expedited. The selected samples of the employed-bees are random at the onlooker stage of the bees and there is a possibility of another search at 25 percent. The data of the current optimal solution and the optimal solution features (M) are used to scale the data to give the new positions which are used to substitute 5 percent of the worst positions in case the multimodal functions have numerous optimal solutions. Firefly algorithm is applied in scout bee phase and a novel position is constructed by moving the identical position to a better solution when founded on the distance of better solution. The proposal of the ABCFF algorithm is given as follows.

Artificial Bee Colony Algorithm

One of the comparatively new nature inspired optimization algorithms is the artificial bee colony algorithm (Karaboga et al., 2014), suggested by Karaboga in 2005 and it also mimics the behavior of honeybee swarms. Bee population comprises three groups (employed bees, onlooker bees and scout bees) that collaborate to locate sources of food. The aid of the communication with another bee makes all the working bees find another food source. The bee in employment will learn its new, better place whether a new better place has been discovered, or not the former place. The ten observer bees decide which information to use on what to explore when it comes to food resources based on the information of the bees employed. The bee that is viewing it will again, encode it when a better place has been located. Upon such a bee having its food source revisited sometime later, it is a scout bee in the subsequent search cycle with its new position. The algorithm ABC can be explained as follows:

(1) Initialization: The initial i^{th} food source $\mathbb{x}_i = (\mathbb{x}_{i,1}, \mathbb{x}_{i,2}, \dots, \mathbb{x}_i, D)$ associated with the i^{th}bee is generated by

$$\mathbb{x}_{i,j} = LB_j + \phi_{i,j}(UB_j - L_j) \quad (4)$$

for $i = 1, 2, 3, \dots,$ NP and $j = 1, 2, 3, \dots, D$ where NP is the number of bees and D is the number of variables or dimension, $\phi_{i,j}$ is a random number in range of (0, 1), and LB_j and UB_j are the lower and upper bounds for the dimension j

(2) Employed Bee Phase: The i^{th} bee shares a piece of information with the $k^{th}h$ bee and generates a new food source VFS_i by using the following equation:

$$VFS_{i,j} = \begin{cases} \mathbb{x}_{i,j}\phi_{i,j}(\mathbb{x}_{i,j} - \mathbb{x}_{k,j}); & j = j^*, \\ \mathbb{x}_{i,j}; & j \neq j^*, \end{cases} \quad (5)$$

in which k is randomly taken between 1 and NP so that $k \neq i, j^*$ is different with ϕ_{ij}, and is randomly taken between 1 and D and phi ij is a random number between [-1 and 1]. It is important to note that VFS_i is not equal to $\mathbb{x}_i$ at the j^{*th} component. The newly possible position VFS i is considered and compared to the previous $\mathbb{x}_i$. When $f(VFS_i) < f(\mathbb{x}_i)$ then, in place of $\mathbb{x}_i$, take VFS_i; otherwise, keep it the same and trial(i) = trial(i) +1 where trial is the number of consecutive unfixed trials.

(3) Onlooker Bee Phase: Onlooker bees choose the food sources according to their quality by means of the probability values p(i), which are calculated by.

$$p(i) = \frac{\text{fitness}(\mathbb{x}_i)}{\sum_{j=1}^{NP} \text{fit}(\mathbb{x}_j)}, \quad (6)$$

Were

$$\text{fitness}(\mathbb{x}_i) = \begin{cases} \frac{1}{1+f(\mathbb{x}_i)}; & f(\mathbb{x}_i) \geq 0,1, \\ 1 + |f(\mathbb{x}_i)|; & \text{otherwise.} \end{cases} \quad (7)$$

and if rand (0, 1) is less than p(i) then generate a new1 (V_i) according to same equation (2). $f(v_i) < f(\mathbb{x}_i)$ then replace $\mathbb{x}_i$ by (v_i); otherwise, leave $\mathbb{x}_i$ and $\text{trial}(i) = \text{trial}(i) + 1$.

(4) Scout bee phase: When the food source $\mathbb{x}_i$ cannot be optimized using the limited number of trials (limit), create a new position of $\mathbb{x}_i$ using (4).

(5) Estimate the optimal position $\mathbb{x}_{best}$ and optimal value f_{best}..

Repeat step (2)-(5) until the stopping criterion has been achieved.

Firefly Algorithm

ABCFF algorithm scout bee stage, firefly algorithm is used and the concept that was introduced by Yang & He, (2013) is outlined as follows. The firefly algorithm (FA) is based on the flashing mechanism of the fireflies whereby the attraction is performed to determine the path that the light stars follow with regard to the strength of the light (brightness) and the attractiveness. The bigger a firefly the more appealing it is i.e. a more appealing firefly than another one attracts the other. Rather, the appeal is directly proportional to the rate of firefly where distance between two fireflies $\mathbb{x}_i$ and $\mathbb{x}_j$ is the Euclidean norm.

$$r_{ij} = \left(\sum_{k=1}^{D} (\mathbb{x}_{ik} - \mathbb{x}_{jk})\right)^{1/2} \quad (8)$$

The attractiveness β_{ij} is computed by

$$\beta_{ij} = \beta_0 e^{-\gamma a t r_{ij}^2} \quad (9)$$

With β_0 signifying the initial attractiveness of light at $atr = 0$ and γ signifying the coefficient of light absorption. For simplicity, β_0=0 and γ=1. By the passage of the firefly $\mathbb{x}_i$ to another more desirable firefly $\mathbb{x}_j$, the new position is presented by

$$\mathbb{x}_i = \mathbb{x}_i + \beta_{ij}(\mathbb{x}_j - \mathbb{x}_i) + (r\ (0,1)\ - \alpha(\ 0.5) \quad (10)$$

Where r (0, 1) and α are the random numbers between 0 and 1.

Proposed ABCFF Method Steps

(1) Initialization: To provide high-quality initial solutions, generate the i^{th} food source , xi, by using the search space division

$$\mathbb{x}_{i,j} = LB_j + \sum_{i=1}^{NP} \sum_{j=1}^{D} \frac{(\phi_{i,j+2i-1})(UB_j - LB_j)}{2i} \quad (11)$$

for $i = 1, 2, 3, \ldots,$ NP and $j = 1, 2, 3, \ldots,$ D, where $\phi_{i,j}$ is a random number in [−1, 1].

(2) Employed Bee Phase: The best position $\mathbb{x}_{best}$ is used to generate a new food source V_i by the following equation:

$$VFS_{i,j} = \begin{cases} \mathbb{x}_{best,j} + \phi_{i,j}(\mathbb{x}_{i,j} - \mathbb{x}_{k,j}); & j = j^*, \\ \mathbb{x}_{i,j}; & j \neq j^*, \end{cases} \quad (12)$$

in which k is randomly taken between 1 and NP in such a way that $k \neq i, j^*$ is randomly taken between 1 and *D* and $\phi_{i,j}$ is a random number in [-1, 1]]. When $f(VFS_i) < f(\mathbb{x}_i)$ replace $\mathbb{x}_i$ by VFS_i, otherwise leave v_i unchanged and set $\text{trial}(i) = \text{trial}(i) + 1$, where trial is the trial number of unimproved trials.

(3) Onlooker Bee Phase: The probability values used to make decisions for onlooker bees are set to a constant $pr(i) = 0.25$. Suppose $\text{rand}(0,1) <$ pri, then (9) is used to generate a new position v_i. If $f(v_i) < f(\mathbb{x}_i)$, then replace $\mathbb{x}_i$ by v_i; otherwise, retain $\mathbb{x}_i$ and set $\text{trial}(i) = \text{trial}(i) + 1$. In addition, the worst positions are replaced by the new ones using the equation.

$$\mathbb{x}_{z_t} M\left[\mathbb{x}_{best} + \phi_t(\mathbb{x}_{z_t} - \mathbb{x}_{r_1}) + \omega_t(\mathbb{x}_{best} - \mathbb{x}_{r_2})\right] \quad (13)$$

Where $z_t, t = 1, 2, \ldots, \lfloor 0.05NP \rfloor$, are the indexes of the worst positions, r_1 and r_2, are the randomly selected indexes in 1 to NP such that the $r_1 \neq r_2 \neq z_t$ are not equal at all times, t, ϕ_t and ω_t are the random numbers in range [-1,1], and M is the best positions acquired by the previous generation.

(4) Scout Bee Phase: Create a new location of the non-revised position $\mathbb{x}_i$ of a scout bee with the aid of the strategy of Firefly algorithm:

$$\mathbb{x}_i = \mathbb{x}_i + e^{-r_{iq}^2}(\mathbb{x}_q - \mathbb{x}_i) + (\text{rand}(0,1) - 0.5) \qquad (14)$$

In which $\mathbb{x}_i$ is the first index that $f(\mathbb{x}_q) < f(\mathbb{x}_i)$, the ABCFF is illustrated using all the steps of the ABC algorithm, and FA is illustrated using Algorithm 1. The agency of employee bee applies the search equation to offer the optimal information resolution to enhance them. The onlooker bee takes 25 percent of the employee bees to make additional moves in search and 5 percent of the worst search positions are replaced in the new position to give the best solution as B. The scout bee stage uses the firefly algorithm to look at an alternative location by substituting locations that were not revised with the aid of the distance procedure to discover the most promising solution.

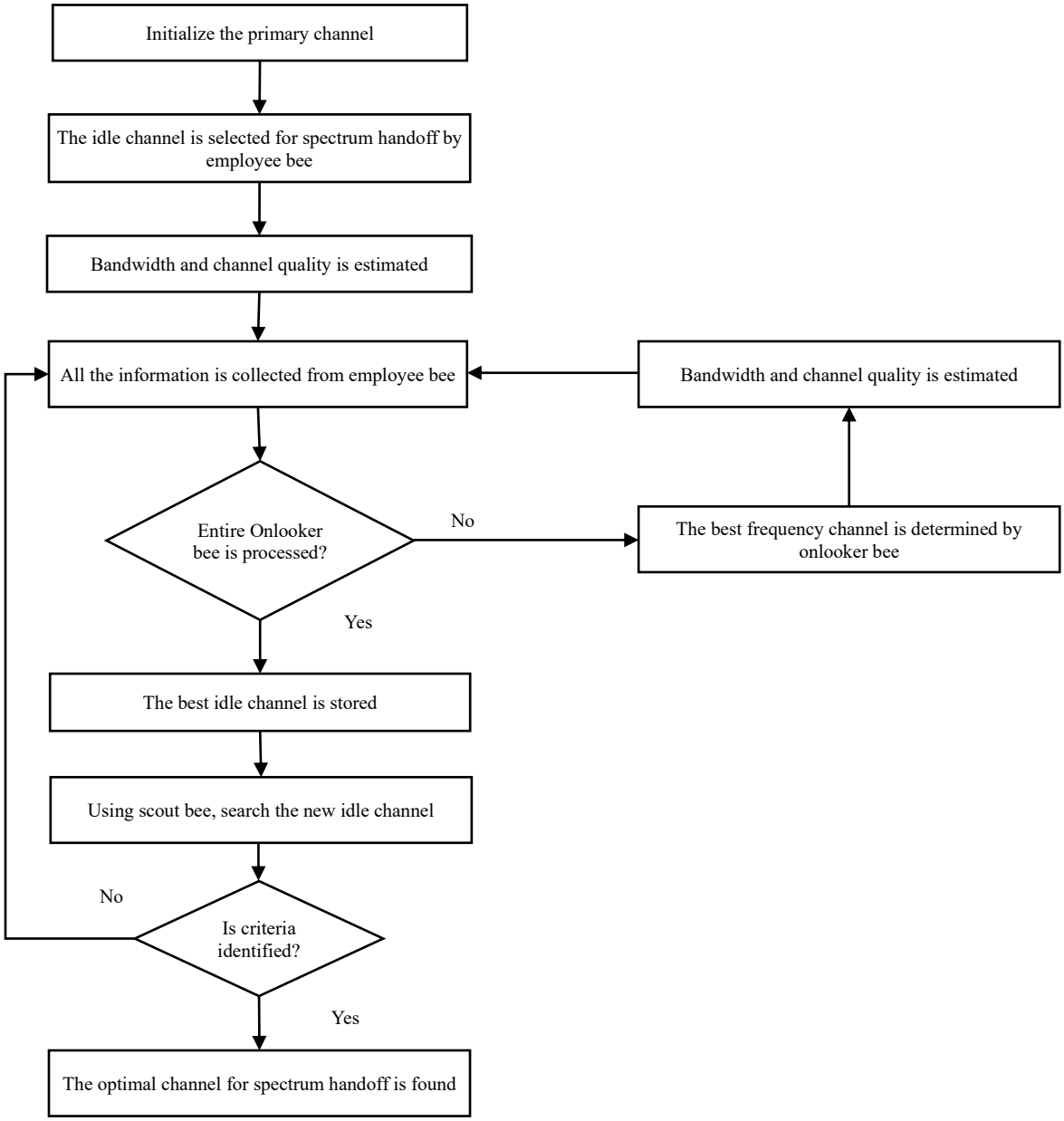

Fig. 2: Proposed ABCFF-Based Feature Selection

The proposed architecture in the form of a diagram is presented in fig. 5, which illustrates the entrance of the main user. The metaheuristic spectrum-sensing algorithm is used to detect the features during the handoff of the secondary user. The optimal features of the spectrum are generated to be used in a cognitive radio network using this proposed algorithm of ABCFF. The steps of calculation of ABCFF are shown in fig. 2.

1. Sensing the idle features information to the active secondary user. The employee bee initiates perceiving the frequency characteristics, and in case it is identified, a new bandwidth is taken into account.
2. The onlooker bee is positioned in the hive and receives the information of the employee bee. Following this information, optimal features are chosen with the help of a fitness value of the spectrum handoff of the secondary user.
3. The scout bee is taken to detect all the features and recalls the busy features and the idle features. In case the features (idle) are known, the previous information about the busy features is erased in the memory.

Algorithm1. Proposed ABCFF Algorithm

1. Fixed parameter: NB, Maxiter, count, number of data features
2. Apply Eq. (4) to set the food source and estimate it. Set t=0.
3. While t<Maxiter
4. // Employee bee phase
5. For the entire employee bee, Eq. (5) is substituted and compared to identify either new or previous features.
6. Calculate the fitness function value and probability values ( )before using (6) and Eq. (7).
7. // Onlooker bee phase
8. For the onlooker bee, the probability value $= 0.25$ & $Zt = 0.05$ is collected from the employee bee, and the B value is considered the best.
9. When Eq. (7) is compared with $r(0,1)$, some new features are observed.
10. Stores the best values that are found in memory.
11. // Scout bee phase
12. Decide the solution using Eq. (7) until Maxiter, and store the best optimal value by replacing the previous value in memory.
13. Find new, updated, and best as final optimal features using Eq. (10)
14. T=t+1
15. Till t=Maxiter.

Hybrid PSO-CNN for KD Classification

The proposed architecture comprises six phases that shed light on the system's actual operation, as shown in fig. 3. The detailed description of each phase is presented next.

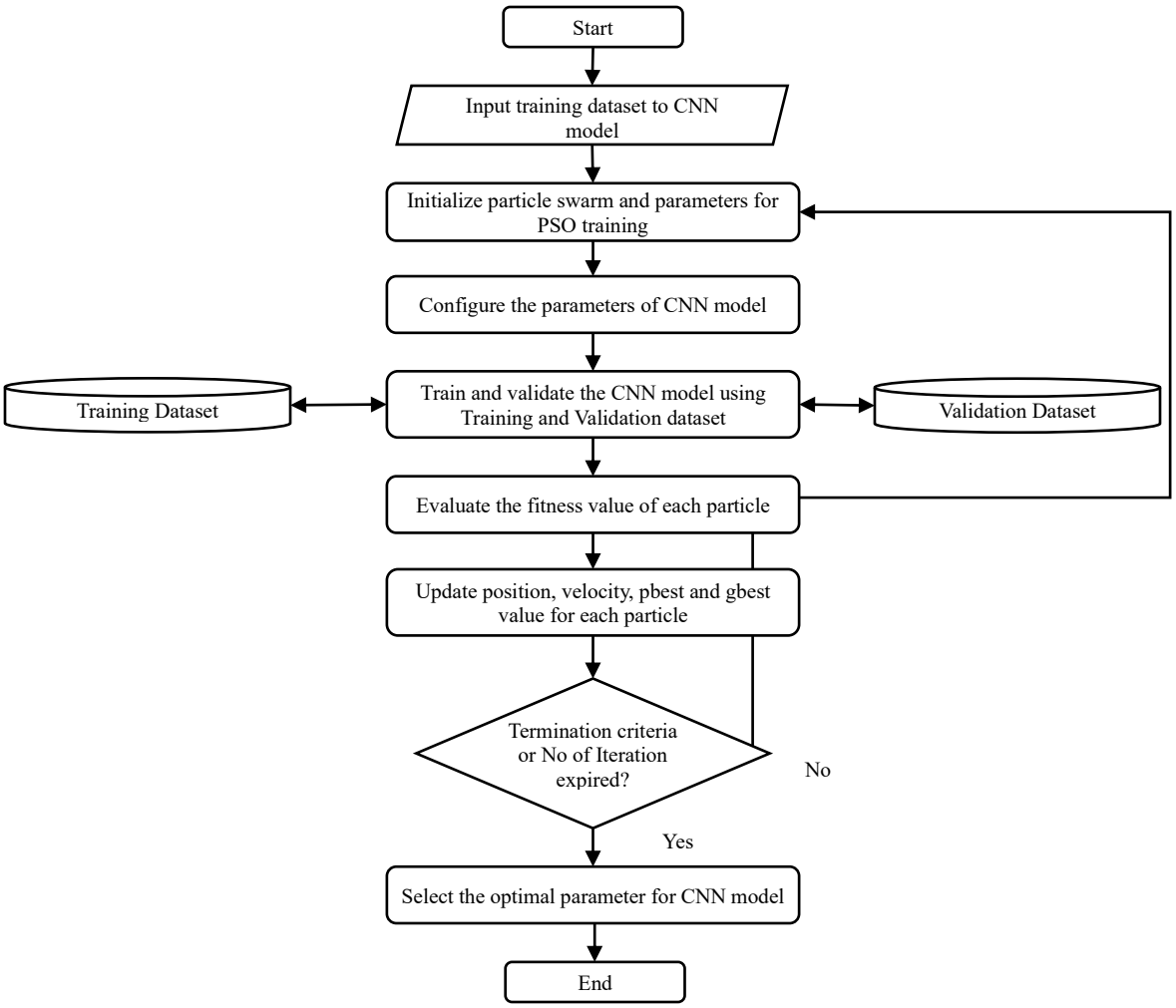

Fig. 3. Flowchart of PSO-CNN

This section outlines the proposed architecture, each layer composing it in detail, and how it has been optimized. It also provides a theoretical background for the 1-D convolutional neural network (CNN) that underpins the proposed architecture. CNNs are just like ANNs that are composed of self-optimizing neurons that are trained to accomplish a given task. This decade has seen the emergence of a 1-D CNN capable of processing 1D data, or time series, for IDS (Azizjon et al., 2020). The architecture proposed based on this 1D CNN concept is depicted in fig. 4 below.

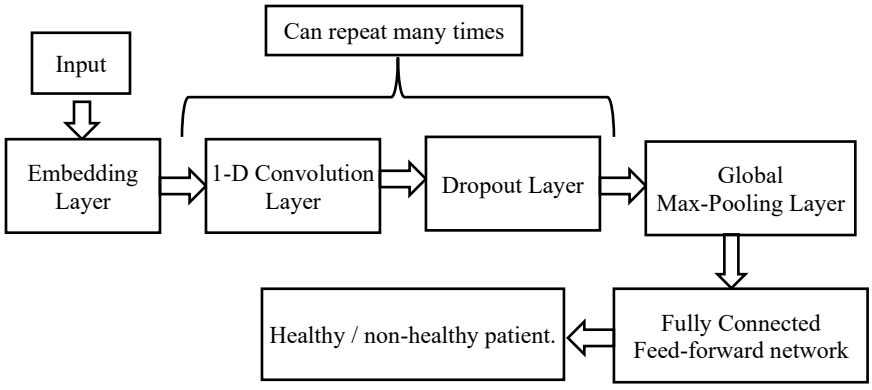

Fig. 4: Architecture of CNN

The 13 important features serving as inputs to the architecture will be utilized as the features of heart diseases which may be classified. The embedding layer then changes these features into a new format, referred to as, a word

embedding. It resembles the bag of words concept of a text data. It will enable one to model the data in a better way, taking into consideration the personal values within each feature. The 1D CNN is linked to the embedding layer, where features are extracted and the 1D CNN is a similar structure to a standard 2D CNN, except that the convolutional operation acts in one direction only that is why it produces a shallow structure and can be easily trained on a regular CPU or embedded development board. The convolution process is useful in isolating useful data features in hierarchies which are useful in classification since they help the model learn patterns that will distinguish between classes in the data set. The size of the output features of a 1D CNN can be estimated using the help of the following equation:

$$\mathbb{x} = \frac{w+2pad-\mathbb{f}}{s} + 1 \quad (15)$$

Where $\mathbb{x}$ is the dimension of the features of the output and w is the dimension of the features of the input $\mathbb{f}$ represents the size of the filter utilized in the convictions, pad is padding which is the value added to the edge prior to the implementation of the convolution process and s is the stride which is the distance moved after the convolution process is implemented. The 1D convolution type does not qualify as a non-linear operation that can be applied to non-linear data in order to classify it. The majority of data in the real world are nonlinear and thus it requires some nonlinear processing after the convolution. Such nonlinear operation is called an activation function. Activation functions are usually the sigmoid activation functions, hyperbolic tangent activation functions, and the rectified linear unit (ReLU). The architecture suggested employs the activation function which is ReLU and is computationally simple, making it possible to compute it at a lower rate. Neither does it have vanishing or blow-up gradients.

This architecture may be multiple layers of convolution with an activation function. The proposed architecture will consist of two 1-D convolutional layers with 128 filters and 3-filter size. Output of the last convolutional layer is then pushed through global max pooling layer which is a pooling operation that takes the biggest maximum value between channels and downsizes the output. The pooling layer is fed with its output to a fully connected layer of 256 neurons that can extract features that can be used to make a classification. This is the same as the hidden layer of the ANN. The highest layer is the one that gives the chances of a classification. The last layer performs the sigmoid activation function as it directly produces outputs of a probability in binary classification.

The 1D CNN architecture that has been proposed contains about 0.13 million of adjustable parameters that will be updated in training. As can be seen, the general CNN architecture fit the training data overfitting and obtained high training accuracy and low validation accuracy. The dropout method was adopted to eradicate overfitting. The dropout method also chooses the neurons to eliminate randomly in training, hence the different networks can be trained in each iteration. This will assist in avoiding the overdependence on the network on particular neurons. The suggested architecture has included the dropout layer once the layers to be trained are complete. The dropout layer contributed to making the accuracy of training and test relatively similar, which means that the network is trained to match the work with the data they have not previously encountered.

PSO Algorithm

PSO algorithm is a computational (Pezoulas et al., 2021), swarm-inspired, and one of the evolutionary optimization strategies designed to address optimization problems through iterative improvement in the goal solution. In the interest of resolving the issue, it adopts an optimization strategy, treating individual particles as separate answers, denoted by the symbol $x_{i,j}$ in the d-dimensional search space, where i is between $(1, 2, \ldots, n)$ and j is between $(1, 2, \ldots, d)$. Each particle updates its location as well as its speed while scanning the entire search space using some mathematical formulae. Their travel is influenced by their personal optimistic position $(\mathcal{p})$ and global optimistic position (g), which are initialized using (13) and (14). The speed v_i and location $x_i$for an individual particle during its path to the best solution are calculated using (13) and (14). The overview of PSO working is depicted in fig. 5.

$$\mathcal{p}_i^{t-1} = f(x) = \begin{cases} \mathcal{p}_i^t & iff(x_i^{t+1}) > \mathcal{p}_i^t \\ x_i^{t+1}, & ff(x_i^{t+1}) \le \mathcal{p}_i^t \end{cases} \quad (16)$$

$$\mathcal{g}_t = min\{\mathcal{p}_i^t\}, where\ i \in [1,2,\ldots,n]\ and\ n > 1 \quad (17)$$

$$v_{i,j}(t+1) = \mathcal{w} \times v_{i,j}(t) + cg_1 \times rand_1 \times \left(\mathcal{p}_i^{t+1} - x_{i,j}(t)\right) + cg_2 \times rand_2 \times \left(g^t - x_{i,j}(t)\right) \quad (18)$$

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1) \quad (19)$$

Thus, cg_1 and cg_2 denote the cognitive and social acceleration component constants that drive the particle's movement towards $\mathcal{p}$ and $\mathcal{g}$, respectively. In the range [0, 1], uniform constant values $rand_1$ and $rand_2$ are chosen. The particle will converge globally due to the weight factor $\mathcal{w}$.

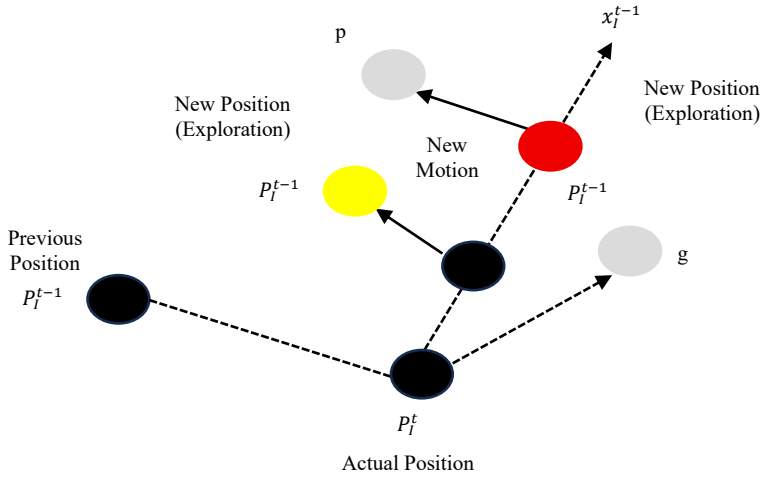

Fig. 5: Overview Of PSO Process

The momentum component, w, plays an important role in PSO, enabling every particle to explore the entire swarm and achieve fast, guaranteed global convergence. Much research was done on it before its use in the PSO working model. This ensured that the particle would explore the entire swarm, both globally and locally, before convergence. Equation (18) calculates the non-linearly decreasing inertia weight.

$$w(linear) = w_{min} + (w_{max} - w_{min}) \times \frac{itr_{max} - itr_i}{itr_{max}} \quad (20)$$

$$w\ (non - linear) = w_{max} \times \exp(-itr) \quad (21)$$

Where w (linear)is a linearly decreasing inertia weight factor, $w_{min} = 0.4$,$w_{max} = 0.9$,itr_{max} is the maximum number of iterations, w (nonlinear) is a non-linearly (exponentially) decreasing inertia weight factor. PSO suffers from one problem: whenever a particle meets a boundary before the convergence step, it doesn't improve its solution further. Still, it attracts other particles towards it, influenced by its $pbest$ and $gbest$ information. This actually prevents all the particles from continuing to improve their optimal locations and from getting stuck after a point due to the untimely meeting at a single point. This scenario can be addressed as an early convergence issue in basic PSO. Hence, this should be avoided for better global convergence. The proposed methodology identifies particles that travel beyond the boundary search areas and brings them back into the search space. This is achieved by controlling the position vector calculation of every updated particle through (20-21) as given below:

$$if\ x_{i,j}(t+1) \leq x_{max} \& if x_{i,j}(t+1) \leq x_{min} \quad (22)$$

$$x_{i,j}(t+1) = x_{i,j}(t+1) \quad (23)$$

$$x_{i,j}(t+1) = x_{i,j}(t+1) - v_{i,j}(t+1) \quad (24)$$

Hyper Parameter Optimization of CNN by PSO

In traditional settings where the size of the dataset is small, the model can be localised in geographic scale to a point where it does not search exhaustively the global search space, and therefore, it is important to carefully tune the hyperparameters so that it does not overfit. Through the assistance of the ground research, PSO chooses 5 parameters to optimize such as number of convolutional layers, number of filters/kernels/convolutional layer, filter/kernel size, pool size, and stride size which will influence CNN performance. Here, 5 parameters are taken into account which are labeled x_1, x_2, x_3, x_4, and x_5 each location of a given particle is associated with a single parameter which is to be optimized with the help of PSO. The specified range of these parameters will be optimized. Fig. 6 is the architecture of a particle in PSO to maximize CNN architecture, and the subsequent are the specifics of the selected parameters with their individual search space in optimization:

1. Let $x1$ = The count of convolutional layers with search space [1, 5]
2. Let $x2$ = The count of filters/kernel in each convolutional layer with search space [32, 128]
3. Let $x3$ = The size of filters/kernel in each convolutional layer with search space [1, 4]
4. Let $x4$ = The size of the pool in each pooling layer with search space [1, 4]
5. Let $x5$= The size of stride [2, 4]

The value of parameters x_3 and x_4 will be set according to the value returned in the range of 1 to 5 by PSO and its related size of filter and pool will be finalized as per below: For example, Let x_3 is the parameter to be discussed then, for $x_3 = 1$, filter size = [3, 3], for $x_3 = 2$, filter size = [5, 5], for $x_3 = 3$, filter size = [7, 7], for $x_3 = 4$, filter size = [9, 9]. The same logic will be applied for $the\ x_4$ parameter, such as for $x_4 = 1$, filter size = [1, 1], for $x_4 = 2$, filter size = [2, 2], for $x_4 = 3$, filter size = [3, 3], for $x_4 = 4$, filter size = [4, 4].

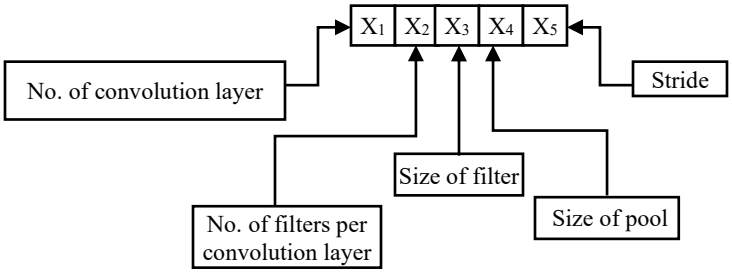

Fig. 6: Hyperparameter Tuning of CNN With Five Variables Using PSO

During the optimization process, $the\ x_1$ parameter plays an important role, as it determines how many convolutional (product-sum) layers are needed to construct a CNN. If PSO outputs the value $x_1 = 1$, then only 1 convolutional layer will be added to the CNN, with the activation of parameters x_2 and x_3 specifying the count and size of filters/kernels for convolution layer 1, and x_4 specifying the size of filters/kernels in the pooling layer. The particle x_5 represents the stride value, which indicates how many pixels it takes to shift the matrix provided. If PSO produces a value of $x_1 = 4$, then 4 convolutional layers will be added to the CNN using the optimal values for x_2 to x_5 obtained by PSO. The counts and sizes of filters/kernels, the sizes of pools, and the strides in each layer will remain (Metre & Sawarkar, n.d.).

Experimental Results and Discussion

The primary aim of the research is to examine the level of the success of proposed machine learning and deep learning classifiers in detecting cardiovascular disease. The experiments in this study were done using the KD dataset. The work utilizes both the machine learning and deep learning approaches. The tests can be run in two ways. Machine learning models will be used to produce better results as the first step (Marini & Walczak, 2015).

In addition, designed an ensemble model based on the use of the ML, which was effective on a specified dataset. The reason behind this is to obtain more credible results. We performed experiments with deep learning methods in the second step. The results of the first are compared to the proposed PSO-CNN and other former algorithms like SOM (Pezoulas et al., 2021), CNN (Xu et al., 2022) and CNN-FCM to identify KD in this section. Due to this new research, it is hoped that these individuals will be identified sooner and in a better way. The overall performance of the classifiers is also measured using statistical measurements as provided in equations (22) to (15).

Precision is defined as the rate of the correctly identified positive findings to the anticipated positive findings.

$$\text{Precision} = \text{TP/TP+FP} \quad (25)$$

A recall the rate of accurately determined positive findings to the findings in a real class.

$$\text{Recall} = \text{TP/TP+FN} \quad (26)$$

F1 measure is the harmonic mean of Precision and Recall, which accounts for false positives and false negatives.

$$\text{F1 measure} = 2*(\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision}) \quad (27)$$

Accuracy is detected as below:

$$\text{Accuracy} = (\text{TP+FP})/(\text{TP+TN+FP+FN}) \quad (28)$$

Here, TP, FP, TN, and FN denote True Positive, False Positive, True Negative, and False Negative, respectively.

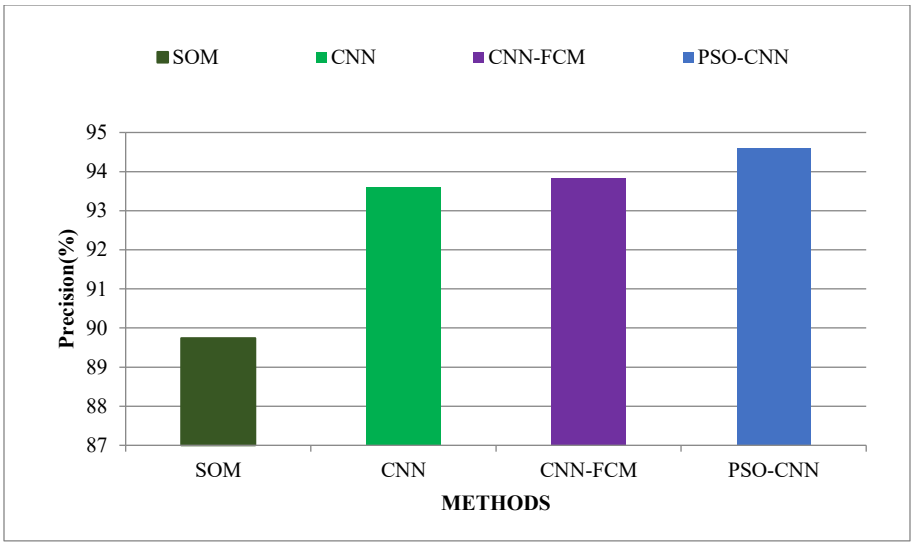

Fig. 7: Precision Outcomes of the Suggested Approach and Traditional Approaches for KD Prediction

Based on the proposed CNN-FCM, the accuracy comparison data would be presented in fig. 7. The findings show that KD can be predicted with the help of the ABCFF feature selection. The amount of features that we can use in the proposed ABCFF does not have a significant effect on the performance of the jointly learned feature transformation. The results of both deep learning systems, as CNN-FCM and PSO-CNN were impressive with the rates of 93.83 and 94.59 respectively. The precision scores of both of the models are nearly identical, albeit lower than the current learning model, due to the fact that the deep learning model was effective on a high volume of data.

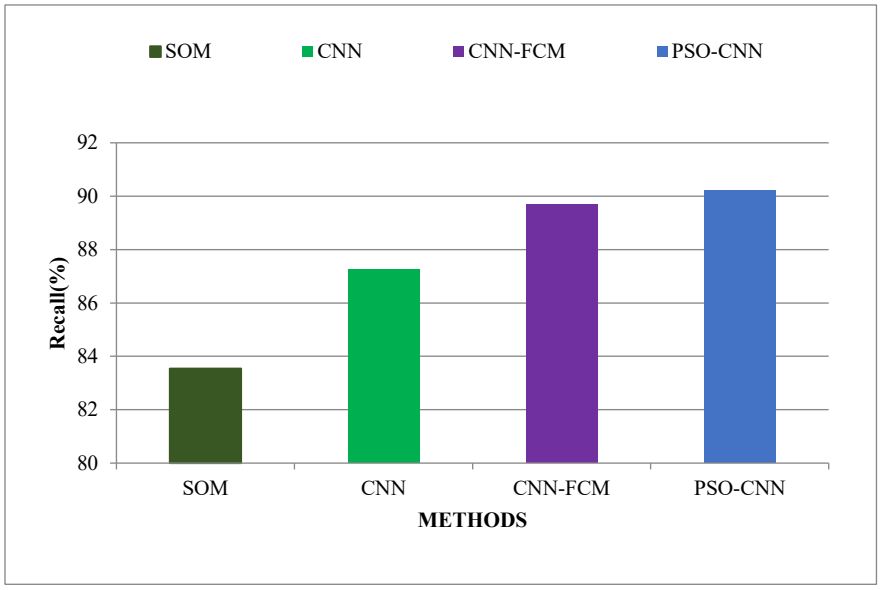

Fig. 8: Recall Comparison Results of the Proposed Approach and Traditional Approaches

The suggested PSO-CNN method's accuracy outcomes are depicted in fig.8. The findings show that the existing methods, such as CNN and SOM, have lower recall percentages of 87.25% and 83.54% than the one presented CNN-FCM, at 89.68%, and the suggested approach has a higher recall percentage of 90.21%. Nonetheless, due to the unavailability of data, the current classifier was not working very well. The findings indicate that the proposed classifier can effectively to identify KD in the KD dataset.

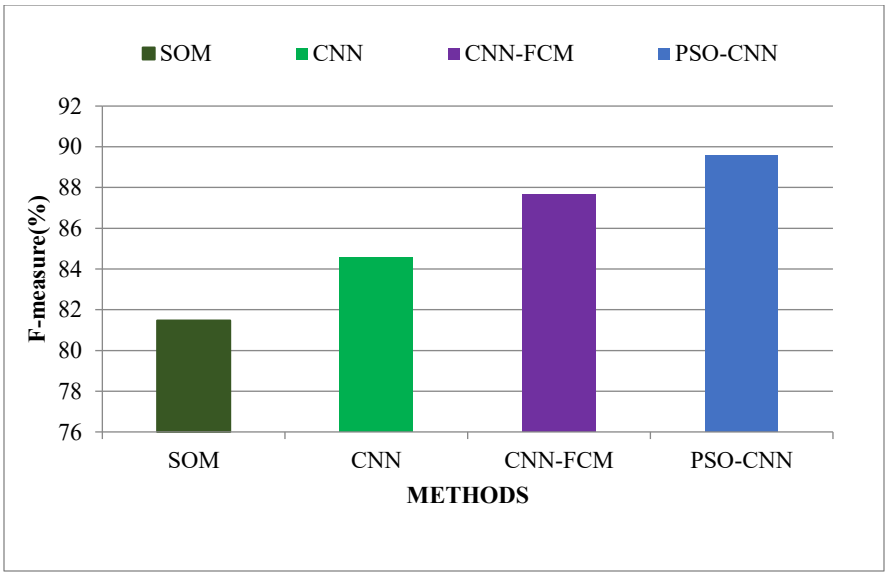

Fig. 9: F-Measure Outcomes of the Suggested Approach and Traditional Approaches for KD Prediction

As shown in fig. 9, the proposed PSO combined with the CNN classifier achieves a high disease prediction rate, significantly superior to those of CNN-FCM, CNN, and SOM. A quantitative study using the F-measure and machine learning yielded results similar to those of the qualitative investigation. According to the KD prediction f-measure, the proposed PSO-CNN is compared to traditional classification methods, achieving a rate of 89.56%. CNNs give a relatively huge amount of features. Therefore, the CNN requires more time to execute as compared to other approaches because it requires numerous feature comparisons. This proposal suggests the ABCFF to minimize the execution time and in turn enhance the f-measure.

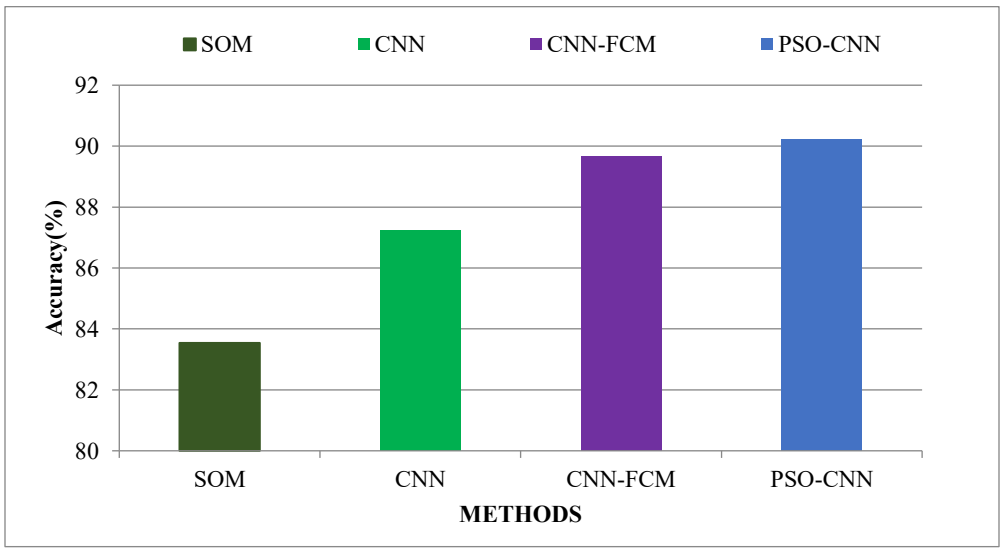

Fig. 10: Accuracy Comparison Outcomes of the Suggested Approach and Traditional Approaches for KD Prediction

The accuracy of the proposed PSO-CNN approach compared to the traditional classification approach is shown in fig. 10. The results indicate that the offered strategy is highly efficient. The success rate of the classifiers will therefore be improved to 90.23% as compared to the current algorithms like CNN-FCM, CNN, and SOM. This is a diverse technique of choosing the optimal characteristics by using CNN+PSO. The KD features are reduced. It was also faster than CNN, as face retrieval is executed with fewer features, which is faster. CNN-PSO needs about 45 seconds to extract features in the dataset.

Conclusion and Future Work

This study suggested the use of hybrid-based deep learning methods to predict KD with PSO-CNN. In our case, we utilised KD data, and the model accuracy was one of the indicators of their performance. Moreover, KNN is applied to impute its data, which affects the performance of the models. The PSO-CNN model, based on the findings, was the most precise when it comes to the prediction of KD in children. ABCF is used in feature extraction. The effective adaptation of ABC algorithm is introduced as known as ABCFF that enhances each step of the ABC algorithm, the initial population will be generated by dividing the search space into smaller parts, the new search equations will be used in the bee employed and onlooker bee phases, the available onlooker bee has the opportunity to find better solutions and certain worst positions will be substituted with the new ones based on the availability of the best position, and the positions that have not been updated will be optimized in the scout bee phase. Lastly, a PSO-based optimization approach using the most appropriate CNN structure to accurately identify and classify the characteristics of diseased children to the corresponding classes, is introduced and tested on one of the largest currently available datasets, to the best of our knowledge. In the future, the identification of the best features in our future work could be done using several strategies. To make a more precise evaluation, one can make use of more data. Lastly, the prediction problem can be addressed with the deep learning and reinforcement learning models to identify cardiovascular disease in a more effective manner.

References

Darby, J. B., & Jackson, J. M. (2021). Kawasaki disease and multisystem inflammatory syndrome in children: an overview and comparison. *American family physician*, *104*(2), 244-252.

Burns, J. C., & Glodé, M. P. (2004). Kawasaki syndrome. *The Lancet*, *364*(9433), 533-544. https://doi.org/10.1016/S0140-6736(04)16814-1

Singh, S., & Kawasaki, T. (2016). Kawasaki disease in India–Lessons learnt over the last 20 years. *Indian pediatrics*, *53*(2), 119-124. https://doi.org/10.1007/s13312-016-0804-5

Newburger, J. W., Takahashi, M., Gerber, M. A., Gewitz, M. H., Tani, L. Y., Burns, J. C., ... & Taubert, K. A. (2004). Diagnosis, treatment, and long-term management of Kawasaki disease: a statement for health professionals from the Committee on Rheumatic Fever, Endocarditis and Kawasaki Disease, Council on Cardiovascular Disease in the Young, American Heart Association. *Circulation*, *110*(17), 2747-2771. https://doi.org/10.1161/01.CIR.0000145143.19711.78

Perotte, A., Ranganath, R., Hirsch, J. S., Blei, D., & Elhadad, N. (2015). Risk prediction for chronic kidney disease progression using heterogeneous electronic health record data and time series analysis. *Journal of the American Medical Informatics Association*, *22*(4), 872-880. https://doi.org/10.1093/jamia/ocv024

Teoh, D. (2018). Towards stroke prediction using electronic health records. *BMC medical informatics and decision making*, *18*(1), 127. https://doi.org/10.1186/s12911-018-0702-y

Jin, B., Che, C., Liu, Z., Zhang, S., Yin, X., & Wei, X. (2018). Predicting the risk of heart failure with EHR sequential data modeling. *Ieee Access*, *6*, 9256-9261.

Yadav, P., Steinbach, M., Kumar, V., & Simon, G. (2018). Mining electronic health records (EHRs) A survey. *ACM Computing Surveys (CSUR)*, *50*(6), 1-40. https://doi.org/10.1145/3127881

Lin, W. C., & Tsai, C. F. (2020). Missing value imputation: a review and analysis of the literature (2006–2017). *Artificial Intelligence Review*, *53*(2), 1487-1509. https://doi.org/10.1007/s10462-019-09709-4

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *nature*, *542*(7639), 115-118. https://doi.org/10.1038/nature21056

Liu, S., Wang, Y., Yang, X., Lei, B., Liu, L., Li, S. X., ... & Wang, T. (2019). Deep learning in medical ultrasound analysis: a review. *Engineering*, *5*(2), 261-275.

Luo, P., & Li, J. (2022). Dual-source computed tomography image information under deep learning algorithm in evaluation of coronary artery lesion in children with Kawasaki disease. *The Journal of Supercomputing*, *78*(9), 11265-11282. https://doi.org/10.1007/s11227-021-04077-9

Benovoy, M., Dionne, A., McCrindle, B. W., Manlhiot, C., Ibrahim, R., & Dahdah, N. (2022). Deep learning-based approach to automatically assess coronary distensibility following Kawasaki disease. *Pediatric cardiology*, *43*(4), 807-815. https://doi.org/10.1007/s00246-021-02790-z

Li, S. M., Liu, W. T., Yang, F., Yi, Q. J., Zhang, S., & Jia, H. L. (2019). Phosphorylated proteomics analysis of human coronary artery endothelial cells stimulated by Kawasaki disease patients serum. *BMC Cardiovascular Disorders*, *19*(1), 21. https://doi.org/10.1186/s12872-018-0982-2

Wen, Y., Wang, X., Guo, Y., Jin, M., Xi, J., Chen, T., ... & Lu, Y. (2021). Predictive value of brachial artery flow-mediated dilation on coronary artery abnormality in acute stage of Kawasaki disease. *Scientific Reports*, *11*(1), 8162. https://doi.org/10.1038/s41598-021-87704-y

Na, J. H., Kim, S., & Eun, L. Y. (2019). Utilization of coronary artery to aorta for the early detection of Kawasaki disease. *Pediatric cardiology*, *40*(3), 461-467. https://doi.org/10.1007/s00246-018-1985-6

Ae, R., Shibata, Y., Kobayashi, T., Kosami, K., Kuwabara, M., Makino, N., ... & Nakamura, Y. (2023). Differences in sensitivity between the Japanese and Z score criteria for detecting coronary artery abnormalities resulting from Kawasaki disease. *Pediatric Cardiology*, *44*(1), 153-160. https://doi.org/10.1007/s00246-022-03008-6

Inagaki, K., Blackshear, C., & Hobbs, C. V. (2019). Deep neck space involvement of Kawasaki disease in the US: a population-based study. *The Journal of Pediatrics*, *215*, 118-122.

Wang, H., Tan, X., Huang, Z., Pan, B., & Tian, J. (2020). Mining incomplete clinical data for the early assessment of Kawasaki disease based on feature clustering and convolutional neural networks. *Artificial Intelligence in Medicine*, *105*, 101859. https://doi.org/10.1016/j.artmed.2020.101859

Lee, H., Eun, Y., Hwang, J. Y., & Eun, L. Y. (2022). Explainable deep learning algorithm for distinguishing incomplete Kawasaki disease by coronary artery lesions on echocardiographic imaging. *Computer Methods and Programs in Biomedicine*, *223*, 106970. https://doi.org/10.1016/j.cmpb.2022.106970

Park, Y. R., Kim, J. J., Yoon, Y. J., Yoon, Y. K., Koo, H. Y., Hong, Y. M., ... & Korean Kawasaki Disease Genetics Consortium. (2016). Establishment of Kawasaki disease database based on metadata standard. *Database*, *2016*, baw109,1-6. https://doi.org/10.1093/database/baw109

Pujianto, U., Wibawa, A. P., & Akbar, M. I. (2019, October). K-nearest neighbor (k-NN) based missing data imputation. In *2019 5th international conference on science in information technology (ICSITech)* (pp. 83-88). IEEE.

Panniem, A., & Puphasuk, P. (2018). A modified artificial bee colony algorithm with firefly algorithm strategy for continuous optimization problems. *Journal of Applied Mathematics*, *2018*(1), 1237823. https://doi.org/10.1155/2018/1237823

Karaboga, D., Gorkemli, B., Ozturk, C., & Karaboga, N. (2014). A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artificial intelligence review*, *42*(1), 21-57. https://doi.org/10.1007/s10462-012-9328-0

Yang, X. S., & He, X. (2013). Firefly algorithm: recent advances and applications. *International journal of swarm intelligence*, *1*(1), 36-50. https://doi.org/10.1504/IJSI.2013.055801

Azizjon, M., Jumabek, A., & Kim, W. (2020, February). 1D CNN based network intrusion detection with normalization on imbalanced data. In *2020 international conference on artificial intelligence in information and communication (ICAIIC)* (pp. 218-224). IEEE.

Marini, F., & Walczak, B. (2015). Particle swarm optimization (PSO). A tutorial. *Chemometrics and intelligent laboratory systems*, *149*, 153-165. https://doi.org/10.1016/j.chemolab.2015.08.020

Metre, V. A., & Sawarkar, S. Particle Swarm Optimization of Convolutional Neural Network for Plant Leaf Disease Identification.

Pezoulas, V. C., Papaloukas, C., Veyssiere, M., Goules, A., Tzioufas, A. G., Soumelis, V., & Fotiadis, D. I. (2021). A computational workflow for the detection of candidate diagnostic biomarkers of Kawasaki disease using time-series gene expression data. *Computational and Structural Biotechnology Journal*, *19*, 3058-3068. https://doi.org/10.1016/j.csbj.2021.05.036

Xu, E., Nemati, S., & Tremoulet, A. H. (2022). A deep convolutional neural network for Kawasaki disease diagnosis. *Scientific Reports*, *12*(1), 11438. https://doi.org/10.1038/s41598-022-15495-x