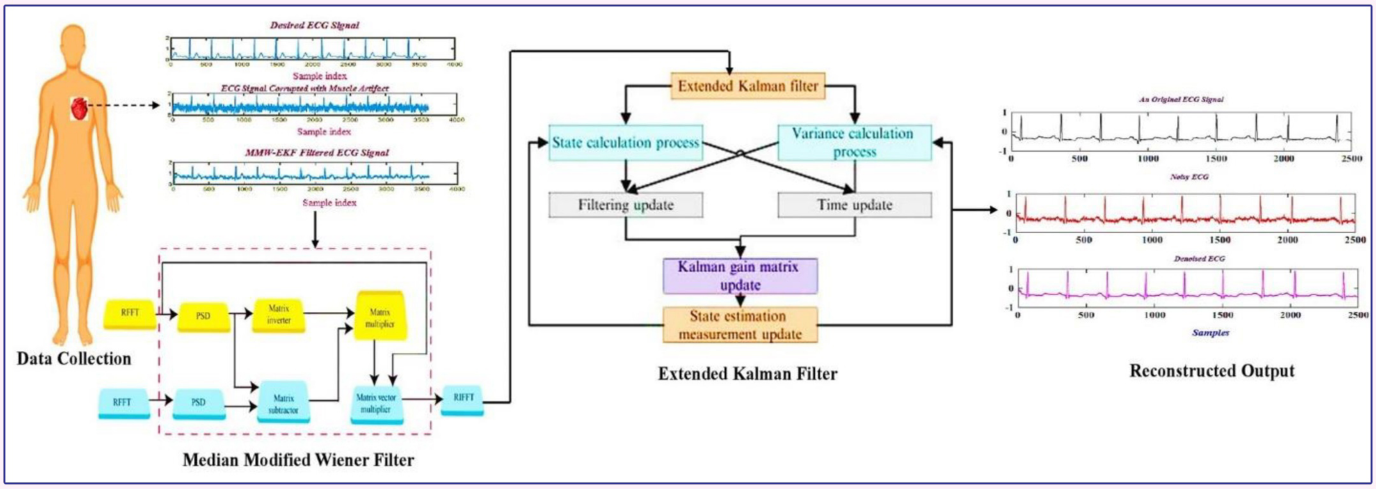
Volume 01, Number 02, 2023

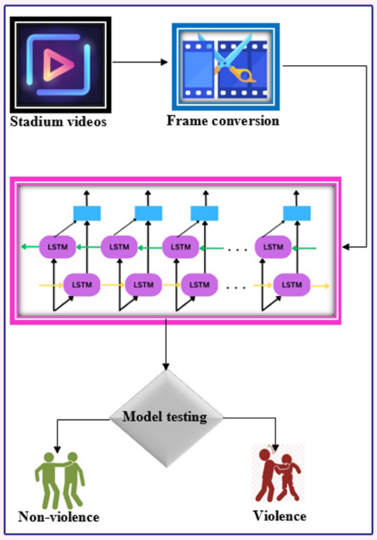
KITS PRESS
For the Research

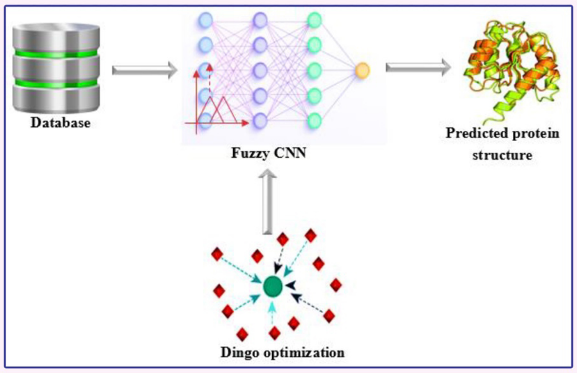
IJDSAI

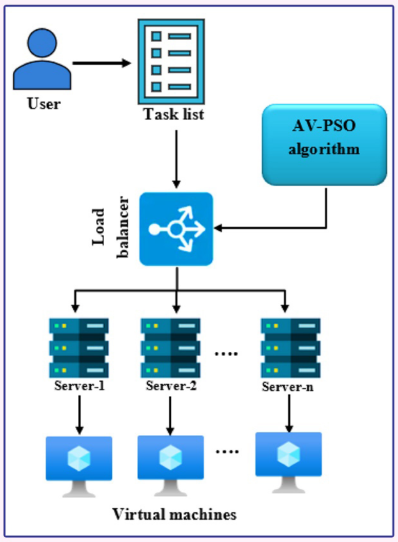
ISSN : 2584-1041

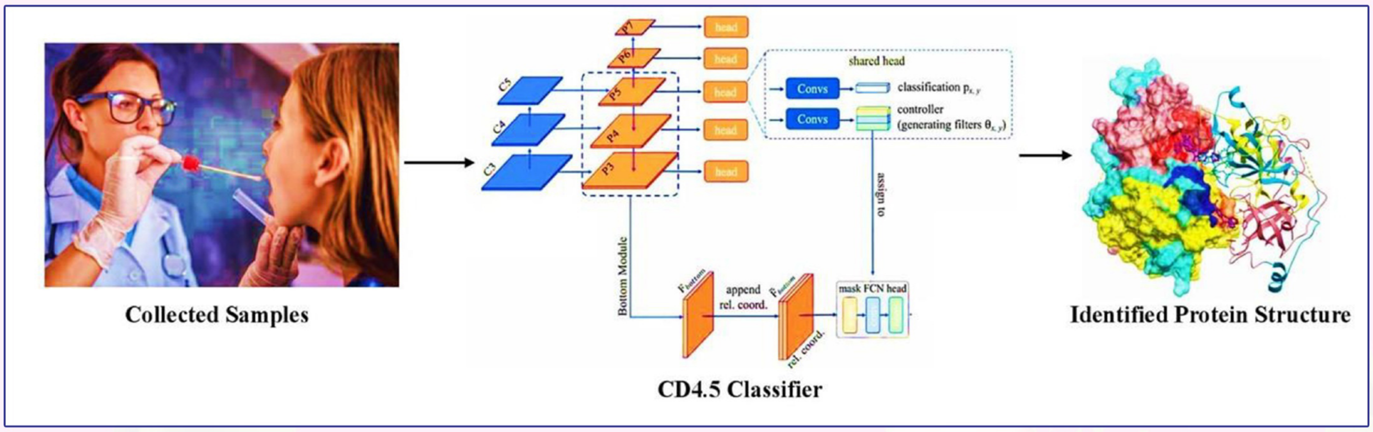
International Journal of Data Science and Artificial Intelligence







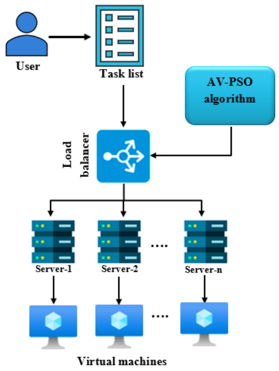




International Journal of Data Science and Artificial Intelligence (IJDSAI)

1. C-AVPSO: DYNAMIC LOAD BALANCING USING AFRICAN VULTURE PARTICLE SWARM OPTIMIZATION

D. Champla, Ghazanfar Ali Safdar, B. Muthukumar and M. Mohamed Sithik

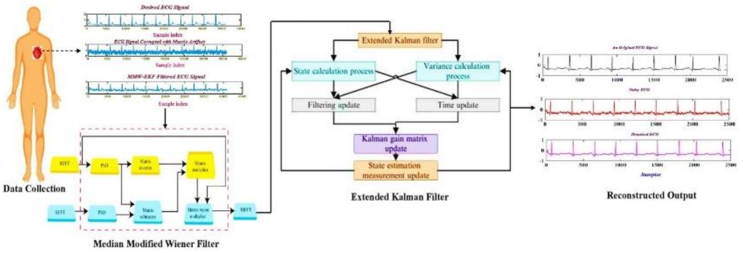


Abstract – Cloud computing is a novel technology that allows consumers to access services from anywhere, at any time, under different conditions, and is controlled by a third-party cloud provider. Cloud task scheduling is a complicated optimisation problem. However, both under- and over-loading conditions cause a range of system problems as far as power consumption, machine failures, and so forth are concerned. consequently, virtual machine (VM) work-load balancing is regarded as a key component of cloud task scheduling. In this paper, a novel cloud-based African vulture particle swarm optimisation [C-AVPSO] has been proposed. Using C-AVPSO, the developed optimization algorithm solves the dynamic load balancing problem effectively. In this method, the exploration space was obtained by using the AVO procedure whereas the enhanced response was identified by the PSO procedure. This algorithm successfully resolves resource utilization, response time, and cost constraints of the task. As a result of combining the AVO and PSO algorithms into the proposed AVPSO algorithm, the convergence rate and performance metrics for load balancing in the cloud environment are improved. To improve the operation's efficiency, the proposed method balances VM loads efficiently. The proposed framework is compared to existing approaches like QMPSO, FIMPSO and ACSO based on energy utilization, degree of imbalance and task migration, response time and resource utilization. The proposed C-AVPSO technique reduces resource utilization of 19.1%, 31%, and 54% than, QMPSO, FIMPSO and ACSO existing techniques.

Keywords – *Cloud computing, Load balancing, C-AVPSO, virtual machine, Optimisation.*

2. A NOVEL INTERNET OF THINGS-BASED ELECTROCARDIOGRAM DENOISING METHOD USING MEDIAN MODIFIED WEINER AND EXTENDED KALMAN FILTERS

L. Jenifer, Xiaochun Cheng, A. Ahilan and P. Josephin Shermila

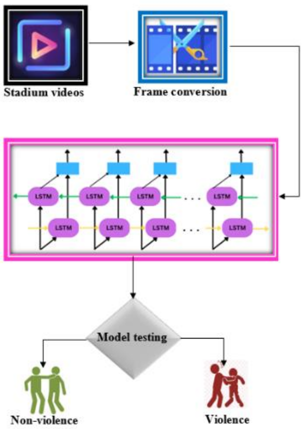

Desired ECG Signal
ECG Signal Corrupted with Muscle Artifact
MMW-EKF Filtered ECG Signal
Sample index
Data Collection
Median Modified Wiener Filter
Extended Kalman filter
State calculation process
Variance calculation process
Filtering update
Time update
Kalman gain matrix update
State estimation measurement update
Extended Kalman Filter
Reconstructed Output

Abstract – The Internet of Things (IoT) offers healthcare applications that benefit customers, physicians, hospitals, and insurance companies. Wearable technology like fitness bands and other wirelessly connected gadgets like blood pressure monitors, blood glucose meters, and heart rate monitors are examples of these uses. The wearable sensor devices utilized in IoT-based Electrocardiogram (ECG) denoising systems continuously produce a huge volume of signals. IoT sensor devices produce ECG signals at a very rapid rate. As a result, the IoT-based health monitoring system generates ECG signals with very high noise levels. A clean ECG signal is needed for effective heart disease management. Imbalanced electrolytes cause an abnormal ECG reading. The noise can also cause fluctuations the ECG signals. This study shows a novel IoT-based ECG denoising method by combining two filters: the Median Modified Weiner (MMW) and the Extended Kalman filter (EKF), to overcome this issue. The characteristic of ECG signals are first subjected to the MMW filter. The extracted ECG signal is then explained with the Extended Kalman filter. MAT LAB simulates the proposed method. Root mean square error (RMSE), contrast-to-noise ratio (CNR), signal contrast, and coefficient of variation (COV) are used in the proposed MMW-EKF framework to the current systems are compared to Signal-to-noise ratio (SNR). We demonstrate how the suggested technique effectively distinguishes between various ECG signals from a noisy sample input.

Keywords – *Electrocardiogram; Internet of things; Denoising; Median Modified Weiner Filter; Extended Kalman Filter.*

3. DETECTION OF VIOLENCE IN FOOTBALL STADIUM THROUGH BIG DATA FRAMEWORK AND DEEP LEARNING APPROACH

M. Dhipa and D. Anitha

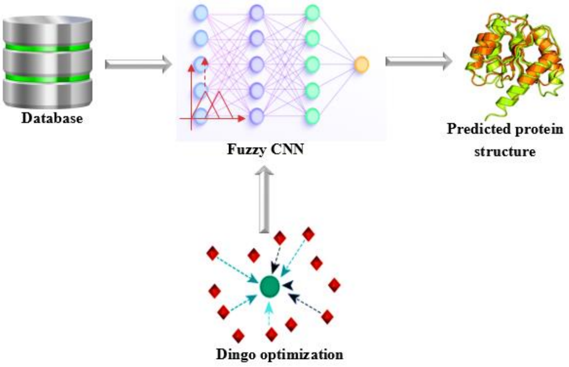


Abstract – Football is the most famous game in the world, with over 4 billion supporters worldwide. Football hooliganism refers to the aggressive or destructive actions of a supporter or player in a stadium while watching or participating in a game. To avoid violence, a real-time violence detection system is required to observe the audience and players behaviour in order to take appropriate action before violence occurs. The input of the system is a massive volume of real-time video feeds from various sources, that are processed using the Flink structure. Using the Histogram of Oriented Gradients (HOG) function in the Flink framework, pictures are partitioned into frames and their characteristics are retrieved. The frames are then labelled on the basis of attributes including Groundside-violence model, Crowdside-violence model, human part model, and Non-violence model, are utilised to train the multihead attention based Bidirectional Long Short-Term Memory network for violent scene detection. The RWF-2000 dataset, which contains the training set (80%) and the test set (20%) was used to train the network and also a dataset comprising 410 video footages with non-violence scenes and 409 video footages with violent situations is created by the videos obtained from a football stadium, to make the algorithm more strong to violence detection. Other existing approaches are used to validate the model's performance. When compared with existing systems, the proposed violence detection methodology significantly increases accuracy upto 1.6453%, precision upto 0.646%, recall upto1.959%, and reduces execution time upto 60% than other existing methods.

Keywords – *Multi-head attention, LSTM, Bidirectional-LSTM, RWF-2000, violence detection, Flink framework.*

4. DINGO OPTIMIZED FUZZY CNN TECHNIQUE FOR EFFICIENT PROTEIN STRUCTURE PREDICTION

P. G. Sreelekshmi and S. C. Ramesh

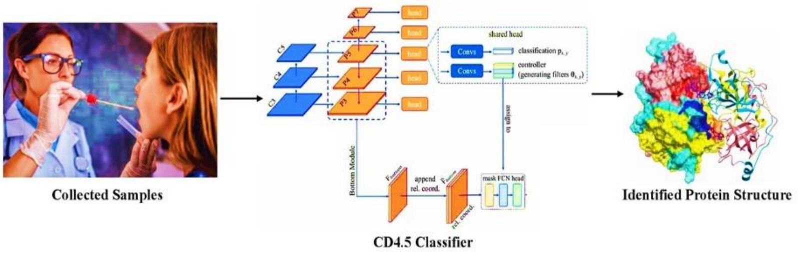


Abstract – Protein is made up of a variety of molecules that are required by living organisms, such as enzymes, hormones, and antibodies. In step 2, the max-pooling layer and the convolutional layer evaluate the input data to create the finest feature map F1, which is half the image size in both horizontal and vertical directions. The full feature is then retrieved in step 2 using the max pooling layer and the residual block at the proper resolution. In this paper, we introduce Di-Fuzzy CNN (Fuzzy Convolutional Neural Network with Dingo optimizer), a novel technique for predicting protein activities that incorporates two types of information they are protein sequence and protein structure. We extract diverse features at different scales utilizing convolutional neural networks to provide comprehensive information for feature segmentation. To handle a variety of uncertainties in feature selection and produce segmentation results that are more dependable, fuzzy logic modules are employed. Finally, we employ Dingo optimization to boost the suggested method's effectiveness and speed in order to produce the best outcomes. Using a variety of datasets, the suggested model has been tested (HSSP, PDB, UGR14b, DSSP). Tests demonstrate that our approach can decrease FPR, increase protein structure accuracy, decrease prediction time, and increase TPR for feature selection. Our predictive model performs better than most state-of-the-art techniques.

Keywords– *Amino Acid Features, Protein Structure, Convolutional Neural Network (CNN), Fuzzy logic, Dingo Optimization algorithm.*

5. IOT-ENABLED PROTEIN STRUCTURE CLASSIFICATION VIA CSA-PSO BASED CD4.5 CLASSIFIER

T. Maris Murugan and A. Jeyam



Abstract – Data mining is a technique for obtaining useful information from vast amounts of information. Big data refers to large amounts of complicated information that is processed, particularly in relation to biological processes. The investigation of protein structures has recently received a lot of attention from structural biologists. The majority of recent research projects have tried to improve protein structure identification in huge data. Feature selection-based protein structure identification in large data analysis, on the other hand, takes a long time. A hybrid crow search algorithm and particle swarm optimization (CSA-PSO) based CD4.5 (CP-CD) approach has been developed to increase Protein Structure Identification accuracy with less amount of time. First samples from the patients are given to IOT-enabled microscope and the details will be stored in big data and then the process will be divided into two steps. At first, feature selection is done using CSA-PSO algorithm, and the classification is done using CD4.5 classifier. This aids in identifying the protein structure and accurately diagnosing the condition, as well as lowering the false positive rate.

Keywords– *Protein structure classification, feature selection, Big data analysis, CD4.5 classifier, IOT-enabled microscope.*

International Journal of Data Science and Artificial Intelligence (IJDSAI)
Volume 01, Issue 02, November – December (2023)

RESEARCH ARTICLE

C-AVPSO: DYNAMIC LOAD BALANCING USING AFRICAN VULTURE PARTICLE SWARM OPTIMIZATION

D. Champla[1, *], Ghazanfar Ali Safdar[2], B. Muthukumar[3] and M. Mohamed Sithik[4]

[1] Department of Computer Science and Engineering, Anna University, Chennai Guindy Mambalam, Tamil Nadu 600025 India.

[2] School of Computing and Information Technology, REVA University, Bengaluru, Karnataka 560064 India.

[3] Department of Computer Science and Engineering, Mohamed Sathak Engineering College, Sathak Nagar, SH 49, Keelakarai, Tamil Nadu, 623806, India.

[4] Department of Computer Networking, University of Bedfordshire, University of Bedfordshire, United Kingdom.

*Corresponding e-mail: *champla.CD32@outlook.com*

Abstract – Cloud computing is a novel technology that allows consumers to access services from anywhere, at any time, under different conditions, and is controlled by a third-party cloud provider. Cloud task scheduling is a complicated optimisation problem. However, both under- and over-loading conditions cause a range of system problems as far as power consumption, machine failures, and so forth are concerned. consequently, virtual machine (VM) work-load balancing is regarded as a key component of cloud task scheduling. In this paper, a novel cloud-based African vulture particle swarm optimisation [C-AVPSO] has been proposed. Using C-AVPSO, the developed optimization algorithm solves the dynamic load balancing problem effectively. In this method, the exploration space was obtained by using the AVO procedure whereas the enhanced response was identified by the PSO procedure. This algorithm successfully resolves resource utilization, response time, and cost constraints of the task. As a result of combining the AVO and PSO algorithms into the proposed AVPSO algorithm, the convergence rate and performance metrics for load balancing in the cloud environment are improved. To improve the operation's efficiency, the proposed method balances VM loads efficiently. The proposed framework is compared to existing approaches like QMPSO, FIMPSO and ACSO based on energy utilization, degree of imbalance and task migration, response time and resource utilization. The proposed C-AVPSO technique reduces resource utilization of 19.1%, 31%, and 54% than, QMPSO, FIMPSO and ACSO existing techniques.

Keywords *– Cloud computing, Load balancing, C-AVPSO, virtual machine, Optimisation.*

1. INTRODUCTION

Cloud computing is the most advanced and rapidly evolving technology in computer science today [27]. The cloud is a network of IT resources, and computing is the act of executing work in remote connection to those resources and charging a pay-as-you-go system. A technology based on the internet that offers a variety of cloud-based services that are efficient, dependable, and inexpensive [29], and these services may be accessed from any device, location, or time. It offers on-demand self-service, which means that when we need a resource, we can use (configure) it without requiring the assistance of a third party. Today, there are various cloud providers available, including Amazon Web Services, Google Cloud, and others. Cloud computing is a computer model that uses the Internet to gather resources [28]. For example, servers, storage, applications, and services.

By ensuring sufficient cloud resource management, the affordable and scalable benefits of cloud computing may be realized. One of the major elements of the cloud structure is that these cloud resources are virtual. Customers can rent services from the Cloud Service Provider (CSP) [30]. With the availability of virtual cloud resources, the CSP's role in providing services to the user is highly complex. As a result, load balancing has received more attention from researchers. This load balancing improves overall system performance. Cloud Service Providers (CSPs) are left with unbalanced computers that have a wide Resources and tasks gradients of user’s consumption as a result [11].

Redistributing workloads as part of a distributed system such as the cloud ensures that no computer is overloaded or underloaded [12,13]. The technique of load balancing has assisted networks and resources in delivering the highest throughput with the quickest reaction times. [14] In load balancing, a number of factors are accelerated to improve the performance of the cloud, such as reaction time, execution time, and system stability [15,16]. Several academics have discussed load balancing strategies, such as (i) static load balancing and (ii) dynamic load balancing, in both heterogeneous and homogeneous situations [17-19].

When the single VM is overloaded with tasks and there are a number of unoccupied VMs in the cloud network, it would be optimal to transfer the tasks from the overloaded

VMs to the underloaded ones [26]. Calculating every conceivable task-resource mapping in a cloud context is challenging, and finding the best mapping is not an easy process [20-24]. Thus, we require an effective task distribution method that can schedule tasks in a way that prevents a large number of virtual computers from being overburdened or underloaded. The cloud task scheduler [25] then begins to perform load balancing operations as soon as it has allocated the task to a virtual machine, so that tasks can be transferred between overloaded and underloaded virtual machines after the task has been allocated to a virtual machine while maintaining the balance of all virtual machines.

This is an overview of this paper's main contributions;

- In this paper, a novel cloud-based African vulture particle swarm optimisation [C-AVPSO] has been proposed.
- The C-AVPSO optimization algorithm efficiently balances load in cloud networks. In this method, the exploration space was obtained using the AVO procedure, while the enhanced response was recognized using the PSO procedure.
- In the developed algorithm, the constraints related to resource utilization, response time, and cost are successfully resolved.
- In a cloud environment, C-AVPSO improves the convergence rate and performance metrics by combining AVO and PSO algorithms. This method maximizes operation efficiency by efficiently balancing VM loads.
- A comparison of the proposed framework and existing approaches like QMPSO, FIMPSO and ACSO is conducted based on energy consumption, degree of imbalance, migration of tasks, response time, and resource usage.

Therefore, the remainder of this article will be structured as follows: The first part of the paper provides a review of the literature. Following that, the proposed research is evaluated in Section 3, results are discussed in Section 4; finally, a conclusion is presented in Section 5.

2. LITERATURE SURVEY

For load balancing in CC, a variety of heuristics and meta-heuristic methods have been used. This section summarises the pertinent work in these areas with a particular emphasis on African vulture particle swarm optimisation (AVPSO) for dynamic load balancing. In this part, we've talked about a few of those technique.

In 2020 Mishra, et al., [1] proposed a category of cloud load balancing algorithms. Distinct ways of load balancing in various cloud computing systems are also described. Load balancing is the process of identifying underloaded and overloaded nodes and the balancing load between them. The simulation is performed in clouds simulator to examine the Heuristic-based performance methods, and the results are detailed.

In 2019 Afzal and Kavitha, [2] proposed a comprehensive encyclopaedic analysis about the load balancing techniques. The benefits and the disadvantages of the present techniques are outlined, and significant issues in developing effective load balancing algorithms are addressed. As a result, 80% of works do not analyze how the load balancing algorithm performs while evaluating performance.

In 2018 Volkova, et al., [3] proposed the cloud analyst analytical tool is used to assess various algorithms. A comparison of algorithm load balancing algorithms is also performed. Load balancing helps the centralized server run better. The load balancing algorithm investigated. Results were compared using Data on total response time, center time, and data center load and processing on an hourly basis cost.

In 2022 Jena, et al., [4] proposed QMPSO is a revolutionary methodology for dynamic load balancing across virtual machines that uses a mixture of an enhanced Q-learning algorithm and amended Particle Swarm Optimization (MPSO). Hybridization's goal is to improve machine performance by distributing the load among the VMs. The algorithm's resilience was demonstrated by comparing the QMPSO simulation results to the current load balancing and scheduling technique.

In 2019 Polepally and Shahu Chatrapati [5] proposed a load-balancing technique based on constraint measure. Each virtual machine's capacity and load are first computed. The load balancing approach computes and analyses the decision factor for each virtual machine. The suggested load balancing method's performance is compared to those of current load balancing techniques like HDLB, DLB, and HBB-LB for capacity and load estimation parameters.

In 2022 Latchoumi, and Parthiban, [6] proposed to obtain the best resource scheduling in a CC scenario, an innovative Quasi Oppositional Dragonfly Algorithm for Load Balancing (QODA-LB) was developed. The main goal of this strategy is to decrease task execution costs and times while keeping the load distributed evenly across all VMs in the CC system. The simulation results showed superior performance to the leading methods and optimal load balancing efficiency.

In 2020 Devaraj, [7] proposed, Firefly and the Improved Multi-Objective Particle Swarm Optimization (FIMPSO,) as a new load balancing algorithm. According to the simulation results, the FIMPSO algorithm produced the most efficient end with shortest common response time of 13.58ms, the highest CPU utilization of 98%, the highest memory utilization of 93%, the highest reliability of 67%, the highest throughput of 72%, and the highest make span of 148, outperforming all other compared methods.

In 2020 Semmoud, et al., [8] proposed a fresh method of load balancing for cloud computing settings. The recommended approach aims to increase system stability while lowering Makespan and VM idle time. When the VM load surpasses the Starvation Threshold, an adaptive limit, the suggested method restricts task transfer. We compared the STLB algorithm to a load balancing algorithm that was inspired by honey bee behaviour, and we found that the suggested method outperformed in terms of average idle time and the quantity of migrations, the HBB-LB algorithm.

In 2021 Balaji, et al., [9] proposed, a load balancing system that addresses optimisation problems by using the adaptive cat swarm optimisation (ACSO) technique. The effectiveness of the suggested technique is assessed with a variety of value indicators, and its performance is contrasted with that of competing techniques. Our suggested solution takes the least time and energy in compared to the current algorithm.

In 2017, Kumar, and Sharma, [10] proposed a load-dynamic balancing method that speeds up cloud resource use while decreasing make-span time. a conventional approach using Cloud load balancing through task migration. In comparison to FCFS and SJF approaches, the experimental findings demonstrate that the suggested technique decreases the manufacture span time and boosts the average resource utilisation ratio.

It can be seen from the reviews above that these methods have some shortcomings. This research proposes a AVPSO technique for dynamic load balancing to address these disadvantages.

3. PROPOSED METHODOLOGY

This article presents a new algorithm for African vulture particle swarm optimization that takes into account the cost, response time, and resource utilization in order to optimize dynamic load balancing. Using this method, you can increase the throughput of your virtual machines, distribute the load among the virtual machines, and maintain the balance of task preferences by adjusting the waiting times for complicated tasks.

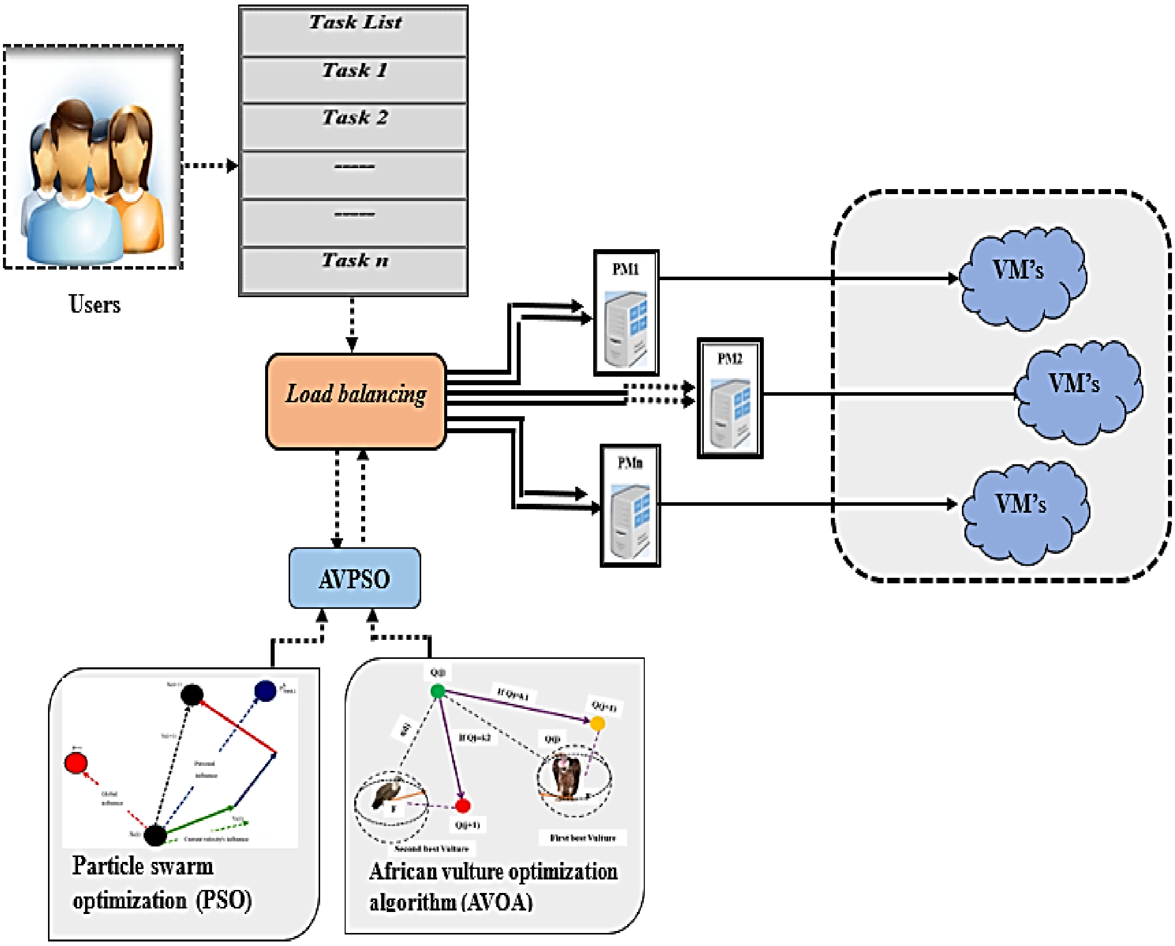


Figure 1. Block diagram of proposed methodology

Fig.1 illustrates the proposed methodology. The primary process in the cloud is workload distribution to virtual machines (VMs). Using node performance data, load balancing decisions are made in the dynamic load balancing mode. Due to low number of VMs, resource delivery to the task is crucial in cloud systems. As a result of the VM being overburdened with jobs, the reaction time of the system is lengthened. Thus, a dynamic load-balancing procedure based on AVPSO is suggested to distribute the tasks across the virtual machines (VMs). This strategy involves moving Underloaded VMs take over tasks from the overloaded VMs. Due to this, the performance and the latent times are sped up. By effectively distributing the load in the cloud, the suggested work-based load balancing system increases resource utilisation while reducing costs and response times. The capacity of each VM is determined after job scheduling. Divide the overflow, underload, and balanced containers according to the VM's remaining capacity. To complete the load balancing procedure, the ideal underload container is found in the proposed task. The tasks are then moved using the migration approach from the worst overloaded VMs to the superior underloaded VM. The AVO algorithm is used into PSO to finish the load balancing process and obtain the search space.

3.1. Load balancing in cloud

Figure 2 depicts the scheduling task in cloud. Each procedure is carried out in the cloud surroundings thanks to cloud computing, which offers cloud facilities to cloud users. Due to the enormous volume of diverse input tasks so as to balance the demands of diverse resources, load balancing is necessary.

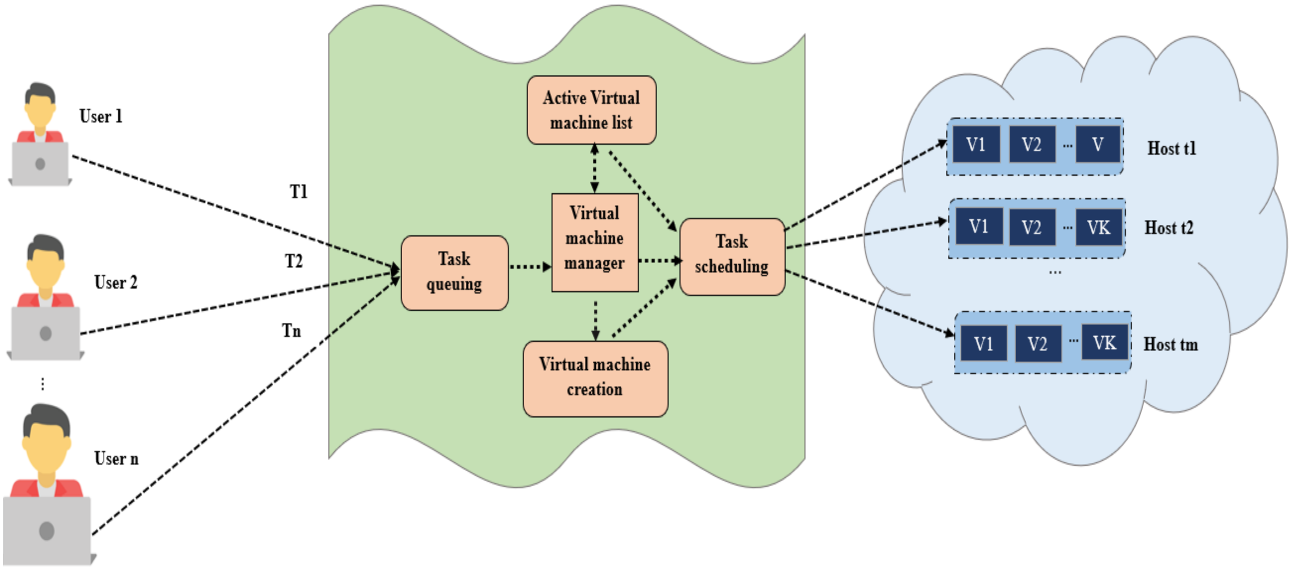


Figure 2. Scheduling cloud tasks for load balancing

The cloud system's task queue receives the tasks with n inputs T_(1),T_2,…T_n .The Virtual Machine head then the get input tasks from task queue and have a comprehensive knowledge of the active VM, existing resources across servers, and the length of the local task queue across all hosts. The system's resource availability was confirmed by the VM manager. The VM management submitted the tasks to the task scheduler if the group of tasks could be completed using the active VMs that are now available. If resource availability does not meet requirements, the VM management generate the necessary VMs in the server. Task allotment in cloud computing is therefore quite difficult. The service's QoS degrades when only a small number of VMs are overloaded, only a small number are free, or when there are fewer tasks to complete. Users may switch to another Cloud provider if they are unhappy with their current service as a result. Each cloud server can only support a certain number of active VMs.

3.2. Particle Swarm Optimization (PSO)

Particle swarm optimisation (PSO), is one of the bio-inspired algorithms, is basic in its quest to find the supreme answer in the problem space. It differs from traditional optimisation techniques in that it only uses the objective function itself and does not depend on the gradient or any differential forms of the objective function. The search is impacted by two distinct learning processes carried out by the particles in PSO. Each particle learns from its own movement-related experiences as well as those of other particles. Learning from one's own experiences is referred to as cognitive learning, whereas social learning involves learning from others. Using social learning, each swarm particle visits the best solution, which is then recorded in each particle's memory as *gbest*. The particle stores the best solution it has independently discovered so far, known as *pbest*, in its memory through cognitive learning. In terms of PSO, time is the iteration. The rate at which the position is changing in relation to the iteration can be regarded of as the velocity in PSO. The iteration counter increases by a factor of unity, which leads to equalize velocity V and position X, the dimensions must be the same.

The most efficient response for a D-dimensional search space, with the i^{th} particle of the swarm at the step time t denoted by a D- dimensional vector., $x_i^t = (x_{i1}^t, x_{i2}^t, \dots . x_{iD}^t)^T$. Likewise, the velocity at step time t can be represented by another D–dimensional vector $v_i^t = (v_{i1}^t, v_{i2}^t, \dots, v_{iD}^t)^T$.The earlier position of the i^{th} particle at the step time t is denoted as $p_i^t = (p_{i1}^t, p_{i2}^t, \dots, p_{iD}^t)^T$. $'g'$indicates which particle is the most efficient in swarm. Using velocity update equation, the i^{th} particle's velocity is upgraded in equation (1)

Velocity update equation:

$$v_{id}^{t+1} = v_{id}^t + c_1 r_1 (p_{id}^t - x_{id}^t) + c_2 r_2 (p_{gd}^t - x_{id}^t) \qquad (1)$$

As shown in (2), the position is upgraded based on the position update equation.

Position update equation:

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \qquad (2)$$

Where, $d = 1, 2, \dots D$ denotes the dimensions and the particle index is represented by $i = 1,2, \dots s$. S is the swarm's size, whereas c1 and c2, or the cognitive and social scaling parameters, are constants. Equations (1) and (2) seem to Indicate that the dimensions of each particle are updated independently. Through the locations of the top places, *gbest* and *pbest*, so far discovered. Equation (1) and (2) outline the

PSO algorithm's fundamental configuration. Algorithm provides a PSO method algorithmic approach.

Algorithm:1

```
Create a D-dimensional swarm which has been initialized with the velocity vectors associated with it;
For $t = 1\ to\ the\ maximum\ bound\ pn\ the\ number\ of\ iterations$ do
    for $i = 1\ to\ S$ do
        for $d = 1\ to\ D$ do
            Apply the velocity update equation 1;
            Apply position update equation 2;
        end
        Compute fitness of updated position;
        If needed, update historical information for pbest and gbest;
    end
    Terminate if gbest meets problem requirements;
end
```

3.3. African Vulture Optimization algorithm (AVOA)

The algorithm, known as the AVOA illustrates how African vultures navigate and forage for food. African vultures are among the unique vultures that can soar to the highest point among the many vulture species. The rotational movements of the African vultures in the sky cause them to fly constantly from one location to another in search of better food supplies. They are at odds with one another in order to obtain the food source. The initial vultures used by the AVO algorithm are some random individuals, and after determining their objective value, their ability is calculated. Each time, one of the top two vultures is either moved or eliminated by a new population type. The following is a list of the prerequisites and points for the regular AVOA.

$$R_i = \begin{cases} Best\ volture\ 1, & if\ pi = L_1 \\ Best\ volture\ 1, & if\ pi = L_2 \end{cases} \tag{3}$$

$$L_1 + L_2 = 1 \tag{4}$$

where,

L1 and L2 define two parameters that are attained before optimisation in the range [0, 1]. to decide which group member is the finest,

$$Pi = \frac{F_i}{\sum_{j=1}^{m} F_i} \tag{5}$$

In equation (5) ‘F’ determines the vultures' level of contentment,

The ratio of vulture starvation has then been determined. As a person runs out of energy, they will engage in combat with nearby, more powerful vultures to obtain food. You can model this as follows:

$$t = k \times \left(sin^{w}\left(\frac{\pi}{2} \times \frac{iter_i}{max_{iter}}\right) + cos\left(\frac{\pi}{2} \times \frac{iter_i}{max_{iter}}\right) - 1 \right) \tag{6}$$

$$F = (2 \times \delta_1 + 1) \times y \times \left(1 - \frac{iter_i}{max_{iter}}\right) + 1 \tag{7}$$

Equation (6) uses w to denote a constant to represent an optimisation procedure, and $iter_i$ to denote the current iteration. y represents a randomly inserted value among 0 and 1. k specifies the random value in range of [2, 2], and δ_1 denotes a random integer between 0 and 1. max_{iter}defines the total number of iterations. The vulture becomes hungry if y decreases to 0, else, it increases to 1.

After that, a random mechanism with two policies was taken into consideration to execute algorithm exploration. The following are examples of how people in an environment hunt for food sources:

If P_1 is less than $rand_{p1}$,

$$P(i+1) = R_i - F + \delta_2 \times ((ub - lb) \times \delta_3 + lb) \tag{8}$$

If P_1 is above or equal to $rand_{p1}$,

$$P(i+1) = R_i - D(i) \times F \tag{9}$$

Where,

$$D(i) = |X \ \times R(i) - P(i)| \tag{10}$$

R denotes a supreme vulture, X indicates how the vulture decides whether or not to keep food acquired from another vulture, which is obtained by $X = 2 \times \delta_i$ where $i = 1,2,3$ two numbers that are created randomly in the value of [0, 1], and $ub\ and\ lb$ denote the boundaries for variables at both lower and higher levels.

Additionally, $|H|$ should be less than 1 in order to abuse the algorithm. This consists of two parts with two siege-fight and rotating flight policies, defined by P_2 and P_3 as two parameters ranging from 0 to 1. Based on the strategy described above, the weakest vulture tries to steal the healthiest food in specified manner that follows;

$$P(i+1) = D(i) \times (F + \delta_4) - d(t) \tag{11}$$

$$d(t) = R_i - P(i) \tag{12}$$

Where, δ_4 is a probability number between 0 and 1.

Moreover, the following is the mathematical description of the vulture's spiral motion:

$$S_1 = R(i) \times \left(\frac{\delta_5 \times P(i)}{2\pi}\right) \times \cos(P(i)) \tag{13}$$

$$S_2 = R(i) \times \left(\frac{\delta_6 \times P(i)}{2\pi}\right) \times \sin(P(i)) \tag{14}$$

$$P(i+1) = R_i - (S_1 + S_2) \tag{15}$$

where δ_5 and δ_6 represent two random numbers between "0" and "1." Most vultures will struggle for food in the beginning if δ_{p_3} ,is a random number between 0 and 1, it’s bigger than (or equal to) P_3. The harsh siege-fight policy has been used if δ_{p_3} is less than P_3. When vultures are famished, it can create a huge competition among them to locate food. The following equation accomplishes this:

$$A_1 = BestVilture_1(i) - \frac{BestVilture_1(i)\ (i) \times P(i)}{BestVilture_1(i)\ (i) - P(i)^2} \times F \tag{16}$$

$$A_2 = BestVilture_2(i) - \frac{BestVilture_2(i)\ (i) \times P(i)}{BestVilture_2(i)\ (i) - P(i)^2} \times F \tag{17}$$

where, $BestVilture_1(i)$ and $BestVilture_2(i)$ represent the best vultures from both sets, while P(i) represents the vector's position in the moment.

$$P(i+1) = \frac{A_1 + A_2}{2} \tag{18}$$

The once-healthy vultures lose their strength and capacity to speak in front of crowds. They then fly to a different location to acquire food once more,

$$P(i+1) = R(i) - |d(t)| \times F \times LF(d) \tag{19}$$

where, LF denotes Levy flight (LF) and calculated analytically as follows:

$$LF(x) = \frac{u \times \sigma}{100 \times |v|^2} \tag{20}$$

$$\sigma = \left(\frac{\Gamma(1+P) \times \sin\frac{\pi\rho}{2}}{\Gamma(1+\rho_2 \times \rho \times 2\left(\frac{\rho-1}{2}\right))} \right) \tag{21}$$

Where, ρ denotes the fixed value, while u and v are the arbitrary numbers between 0 and 1.

4. RESULT AND DISCUSSION

This algorithm is implemented in Cloud Sim as a load balancing algorithm based on C-AVPSO. Our proposed method is similar to the traditional methods QMPSO, FIMPSO, ACSO in terms of the energy utilization, degree of im-balance, number of tasks migration, response time and resource utilization.

4.1. Evaluation metrics

4.1.1. Energy utilization

When compared to other algorithms like FIMPSO, ACSO, and QMPSO during load balance, the proposed C-AVOPSO technique used the most energy. Also, the energy utilisation analysis revealed that, when compared to other algorithms, the proposed C-AVPSO approach required the least amount of energy (by altering the number of VMs from 0 to 250). Energy utilization vs number of VM's and number of tasks is shown in Figures 3 and 4.

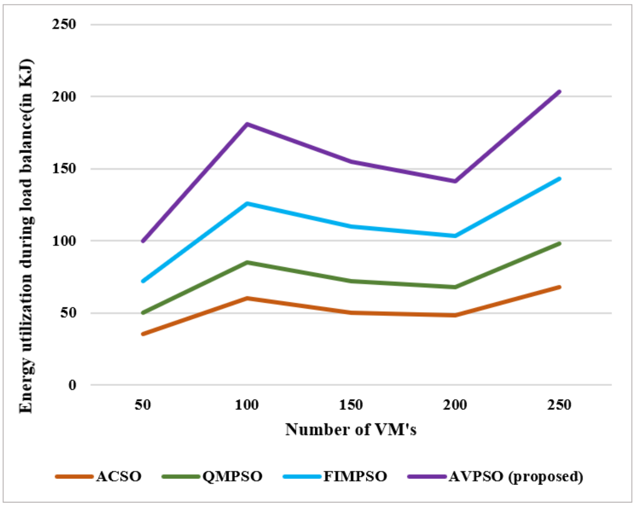


Figure 3. Energy utilization vs number of VM's

By comparing the suggested AVPSO to the various current algorithms, it was discovered that the proposed AVPSO used the most energy within number of Tasks from 100 to 1500.

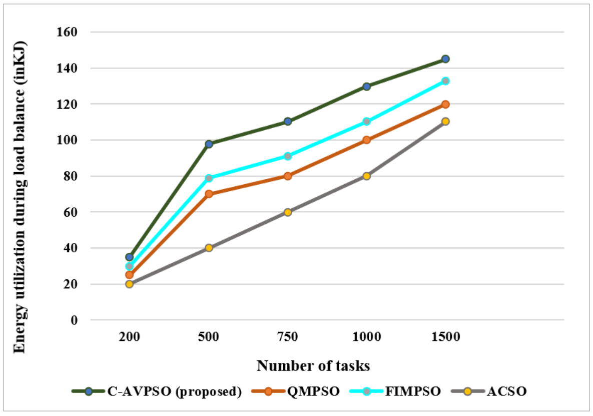


Figure 4. Energy utilization vs number of tasks

4.1.2. Migration

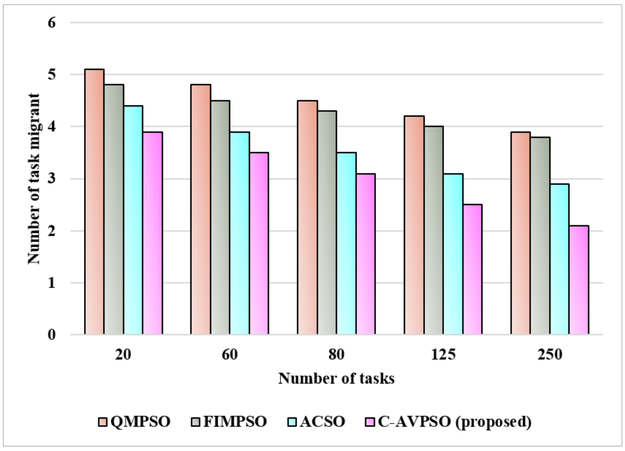


Figure 5. Migration Vs Number of tasks

Figure. 5 illustrates how the number of tasks migrated in relation via total number of tasks. Comparing the C-AVPSO technique to the current QMPSO, FIMPSO, and ACSO algorithms, it was discovered that there was less task migration.

4.1.3. Degree of imbalance

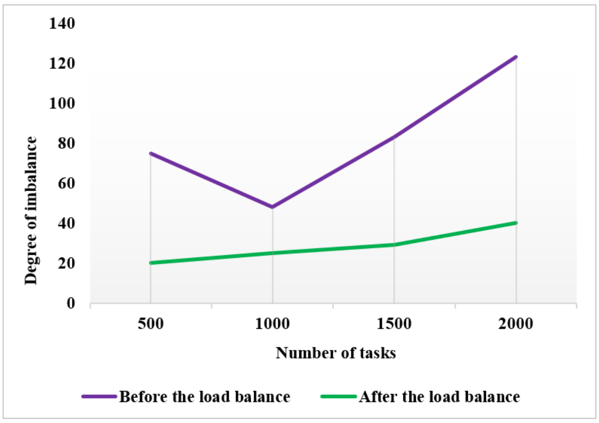


Figure 6. Degree of imbalance Vs Number of tasks

The reduction of imbalance associated with greater load balancing results in the cloud's optimal load balancing. The amount of imbalance determines how long jobs must wait. In general, the load balancing is based on how many jobs the users have requested. The degree of im-balancing after load balancing and before load balancing is depicted in Fig. 6. After the load balancing procedure, it shows that the produced C-AVPSO provide lower degree of imbalance.

4.1.4. Resource utilization

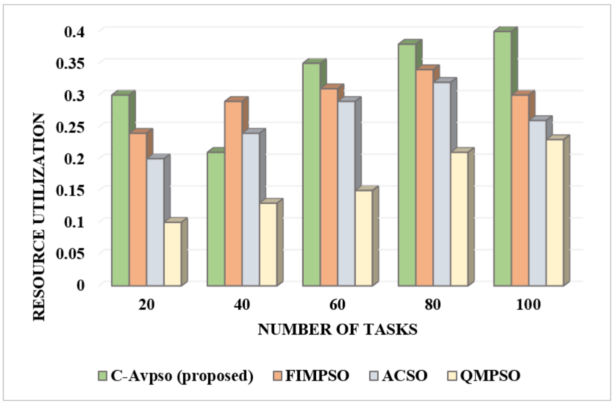


Figure 7. Resource utilization Vs Number of tasks

According to Figure 7, the suggested strategy's average resource utilisation performs admirably in every case when compared to other alternatives. This is because the suggested strategy enables the simultaneous assignment of each individual work to the best processor available. The effectiveness of the suggested strategy can be increased by maximising resource use.

4.1.5. Response time

Figure 8 shows a comparison of response times for various job counts. Between 100 to 500 jobs can be found in both the proposed and current algorithms. When a load balancing system allocates VMs with lower load conditions in response to user demand, this is known as its response time. Comparing the proposed AVPSO to the current QMPSO, FIMPSO, and ACSO approaches, it has the highest response.

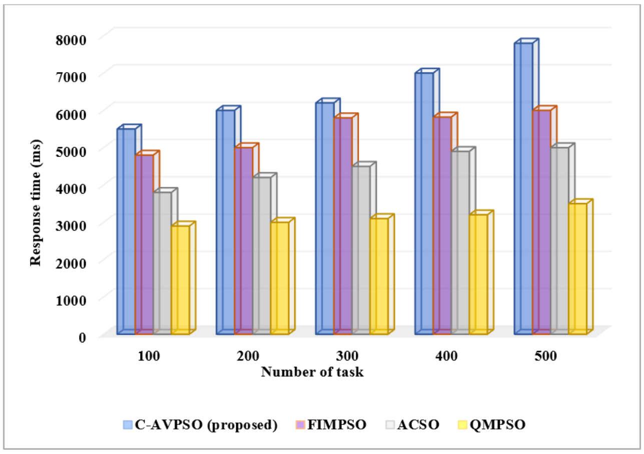


Figure 8. Response time Vs Number of task

5. CONCLUSION

In this paper, a novel cloud-based African vulture particle swarm optimisation [C-AVPSO] has been proposed. Using C-AVPSO, the developed optimization algorithm solves the dynamic load balancing problem effectively. In this method, the exploration space was obtained by using the AVO procedure whereas the enhanced response was identified by the PSO procedure. This algorithm successfully resolves resource utilization, response time, and cost constraints of the task. As a result of combining the AVO and PSO algorithms into the proposed AVPSO algorithm, the convergence rate and performance metrics for load balancing in the cloud environment are improved. To improve the operation's efficiency, the proposed method balances VM loads efficiently. The suggested method was implemented in cloud sim tool. The proposed framework is compared to existing approaches like QMPSO, FIMPSO and ACSO Based on energy utilization, degree of imbalance and task migration, response time and resource utilization. The proposed C-AVPSO technique reduces resource utilization of 19.1%, 31%, and 54% than, QMPSO, FIMPSO and ACSO existing techniques.

CONFLICTS OF INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

FUNDING STATEMENT

Not applicable.

ACKNOWLEDGEMENTS

The author would like to express his heartfelt gratitude to the supervisor for his guidance and unwavering support during this research for his guidance and support.

REFERENCES

[1] S.K. Mishra, B. Sahoo, and P.P. Parida, “Load balancing in cloud computing: a big picture”, *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 2, pp.149-158, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[2] S. Afzal, and G. Kavitha, “Load balancing in cloud computing–A hierarchical taxonomical classification”, *Journal of Cloud Computing*, vol. 8, no. 1, pp. 22, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[3] V.N. Volkova, L.V. Chemenkaya, E.N. Desyatirikova, M. Hajali, A. Khodar, and A. Osama, “Load balancing in cloud computing”, *In 2018 IEEE conference of russian young researchers in electrical and electronic engineering (EIConRus)*, IEEE. pp. 387-390, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[4] U.K. Jena, P.K. Das, and M.R. Kabat, “Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment”, *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 6, pp. 2332-2342, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[5] V. Polepally, and K. Shahu Chatrapati, “Dragonfly optimization and constraint measure-based load balancing in cloud computing”, *Cluster Computing*, 22(Suppl 1), pp. 1099-1111, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[6] T.P. Latchoumi, and L. Parthiban, “Quasi oppositional dragonfly algorithm for load balancing in cloud computing environment”, *Wireless Personal Communications*, vol. 122, no. 3, pp. 2639-2656, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[7] A.F.S. Devaraj, M. Elhoseny, S. Dhanasekaran, E.L. Lydia, and K. Shankar, "Hybridization of firefly and improved multi-objective particle swarm optimization algorithm for energy efficient load balancing in cloud computing environments", *Journal of Parallel and Distributed Computing*, vol. 142, pp. 36-45, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[8] A. Semmoud, M. Hakem, B. Benmammar, and J.C. Charr, "Load balancing in cloud computing environments based on adaptive starvation threshold", *Concurrency and Computation: Practice and Experience*, vol. 32, no. 11, pp. e5652, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[9] K. Balaji, P.S. Kiran, and M.S. Kumar, "WITHDRAWN: An energy efficient load balancing on cloud computing using adaptive cat swarm optimization", 2021. [CrossRef] [Google Scholar] [Publisher Link]

[10] M. Kumar, and S.C. Sharma, "Dynamic load balancing algorithm for balancing the workload among virtual machine in cloud computing", *Procedia computer science*, vol. 115, pp. 322-329, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[11] S. Afzal, and G. Kavitha, "Optimization of task migration cost in infrastructure cloud computing using IMDLB algorithm", *In: 2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET)*, pp. 1–6, (2018, December). [CrossRef] [Google Scholar] [Publisher Link]

[12] R. Achar, P.S. Thilagam, N. Soans, P.V. Vikyath, S. Rao, A.M. Vijeth, "Load balancing in cloud based on live migration of virtual machines", *In: 2013 annual IEEE India Conference (INDICON)*, pp. 1–5. (2013, December). [CrossRef] [Google Scholar] [Publisher Link]

[13] D. Magalhães, R.N. Calheiros, R. Buyya, D.G. Gomes, "Workload modeling for resource usage analysis and simulation in cloud computing", *Comp Elect Eng*, vol. 47, pp. 69–81, (2015). [CrossRef] [Google Scholar] [Publisher Link]

[14] R.N. Calheiros, R. Ranjan, C.A. De Rose, and R. Buyya, "Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services". arXiv preprint arXiv:0903.2525. 2009. [CrossRef] [Google Scholar] [Publisher Link]

[15] S. Dam, G. Mandal, K. Dasgupta, P. Dutta, "Genetic algorithm and gravitational. emulation-based hybrid loads balancing strategy in cloud computing", *In: Proceedings of the 2015 third international conference on computer, communication, control and information technology (C3IT)*, pp. 1–7, 2015, February. [CrossRef] [Google Scholar] [Publisher Link]

[16] A. Dave, B. Patel, G. Bhatt, "Load balancing in cloud computing using optimization techniques: a study", *In: International Conference on Communication and Electronics Systems (ICCES)*, pp. 1–6, 2016, October. [CrossRef] [Google Scholar] [Publisher Link]

[17] D.A. Shafiq, N.Z. Jhanjhi, and A. Abdullah, "Load balancing techniques in cloud computing environment: A review", *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 7, pp. 3910-3933, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[18] J.C. Bansal, "Particle swarm optimization", *Evolutionary and swarm intelligence algorithms*, pp.11-23, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[19] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview", *Soft computing*, vol. 22, pp. 387-408, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[20] M. Farid, R. Latip, M. Hussin, and N.A.W. Abdul Hamid, "A survey on QoS requirements based on particle swarm optimization scheduling techniques for workflow scheduling in cloud computing", *Symmetry*, vol. 12, no. 4, pp. 551, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[21] D. Wu, "Cloud computing task scheduling policy based on improved particle swarm optimization", *In 2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS)*, pp. 99-101, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[22] Y. Wang, S. Li, H. Sun, C. Huang, and N. Youssefi, "The utilization of adaptive African vulture optimizer for optimal parameter identification of SOFC", *Energy Reports*, vol. 8, pp. 551-560, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[23] B. Abdollahzadeh, F.S. Gharehchopogh, and S. Mirjalili, "African vulture's optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems", *Computers & Industrial Engineering*, vol. 158, pp.107408, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[24] I.M. Ibrahim, "Task scheduling algorithms in cloud computing: A review", *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 4, pp. 1041-1053, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[25] E.H. Houssein, A.G. Gad, Y.M. Wazery, and P.N. Suganthan, "Task scheduling in cloud computing based on meta-heuristics: review, taxonomy, open challenges, and future trends", *Swarm and Evolutionary Computation*, vol. 62, pp. 100841, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[26] M. Masdari, and M. Zangakani, "Green cloud computing using proactive virtual machine placement: challenges and issues", *Journal of Grid Computing*, vol. 18, no. 4, pp. 727-759, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[27] P. Srivastava, and R. Khan, "A review paper on cloud computing", *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 8, no. 6, pp. 17-20, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[28] M.I. Malik, S.H. Wani, and A. Rashid, "Cloud Computing-Technologies", *International Journal of Advanced Research in Computer Science*, vol. 9, no. 2, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[29] K.D. Patel, and T.M. Bhalodia, "An efficient dynamic load balancing algorithm for virtual machine in cloud computing", *In 2019 International conference on intelligent computing and control systems (ICCS) IEEE*, pp. 145-150, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[30] A. Rashid, and A. Chaturvedi, "Cloud computing characteristics and services: a brief review", *International Journal of Computer Sciences and Engineering*, vol. 7, no.2, pp. 421-426, 2019. [CrossRef] [Google Scholar] [Publisher Link]

AUTHORS

D. Champla, currently doing as a Research Scholar, Department of Computer Science and Engineering, Anna University, Chennai, SRM Easwari Engineering college, Chennai. I have experience of over 4 years in the field of engineering, academics, administration and active research. I am an alumnus of JNTU Hyderabad. I was completed my previous Degrees, "B tech" Vidhya Bharathi Institute of Technology in 2011, "M tech" Ashoka Institute of Engineering and Technology in 2014, Affiliated to JNTU Hyderabad.

Ghazanfar Ali Safdar received a BSc (Hons) in Electrical Engineering from University of Engineering and Technology, Pakistan in 1997. He subsequently received M. Eng in Computer Science and Telecommunications from University of Technology, Grenoble, France in 2001.He then worked from 2005-2008 with the same university as a Research Fellow involved in the security and authentication protocols for wireless networks. Ghazanfar Ali is currently working as a lecturer in Computer Networking at University of Bedfordshire. He has also worked as R & D engineer with Carrier Telephone Industries (SIEMENS), Pakistan and Schlumberger, France. Dr Ghazanfar Ali Safdar, has been selected to be an associate editor for IET Networks, the journal of the prestigious Institution of Engineering & Technology.

B. Muthu Kumar, Professor, School of Computing and Information Technology, REVA University, Bengaluru received his B. E (CSE) degree from Anna University, Chennai in the year 2005, M.Tech (CSE) (Gold Medalist) received from Dr. MGR. University, Chennai in the year 2007 and Doctoral degree from St. Peter's University, Chennai in the year 2013. He is having more than 16 years of teaching experience in reputed engineering colleges. He has published more than 40 peer reviewed International Journals, 50 International/National Conference and attended more than 150 Workshops/FDPs/Seminars etc., He organized many events like Conference/FDPs/Workshops/Seminars/Guest Lecture. He has published more than 10 patents in various fields like Wireless Sensor Networking, Image Processing, Optimization Techniques and IoT. He received nearly 5.67 Lakhs funding from various agencies like AICTE, ATAL and IEI. He has written 2 books from reputed publishers. He received Best Researcher Award in the year 2021 and Innovative Research and Dedicated Professor Award in Computer Science and Engineering in the year 2018. He has professional membership on ISTE, CSI, IEI, IACSIT, IAENG, CSTA, and SIAM. He has invited as Guest Lecture / Chairperson / Examiner / Reviewer/Editorial Board Member in various Institutions/Journals/Conferences. He is a recognized supervisor in Anna University, Chennai and currently guiding 4 research scholars. His areas of interest are Image Processing, Wireless Networks, IOT and Computing Techniques.

M. Mohamed Sithik, Associate Professor, Department of Computer Science and Engineering, Mohamed Sathak Engineering College, Kilakarai received his B.E (CSE) degree from Madurai Kamarajar University in the year 2004, and M.E CSE received from Anna University Tiruchirappalli in the year of 2011. He is having more than 18 years of teaching experience in reputed engineering college. He has published more than 5 peer reviewed Journals, 10 International/National Conference and attended more than 100 Workshops/FDPs/Seminars etc., He organized many events like Conference/FDPs/Workshops/Seminars. He has published 2 patents in the area of Artificial Intelligence and Telehealth Technologies. He has written 1 book from reputed publisher. He received Best Techie Award in the year 2022. He has professional membership on IEEE. His current research focuses on Secure RPL and Congestion control in Internet of Things (IoT). His research interests include Data Science, Deep Learning, Load balancing, Energy consumption, Security in IoT devices.

Arrived: 08.11.2023

Accepted: 11.12.2023

International Journal of Data Science and Artificial Intelligence (IJDSAI)
Volume 01, Issue 02, November – December (2023)

RESEARCH ARTICLE

A NOVEL INTERNET OF THINGS-BASED ELECTROCARDIOGRAM DENOISING METHOD USING MEDIAN MODIFIED WEINER AND EXTENDED KALMAN FILTERS

L. Jenifer[1, *], Xiaochun Cheng[2], A. Ahilan[3] and P. Josephin Shermila[4]

[1]Department of Electrical and Electronics Engineering, Sathyabama Institute of Science and Technology, Chennai, 600119, India.
[2]Department of Computer Science, Swanesa University, SA1 8EN Wa & United Kingdom.
[3] Department of Electronics and Communication Engineering, PSN College of Engineering and Technology, Tirunelveli, Tamilnadu, 627 152, India.
[4]Department of Artificial Intelligence and Data Science, R.M.K. College of Engineering and Technology, Chennai, Tamilnadu, India.

[*]Corresponding e-mail: jenifer750l@outlook.com

Abstract – The Internet of Things (IoT) offers healthcare applications that benefit customers, physicians, hospitals, and insurance companies. Wearable technology like fitness bands and other wirelessly connected gadgets like blood pressure monitors, blood glucose meters, and heart rate monitors are examples of these uses. The wearable sensor devices utilized in IoT-based Electrocardiogram (ECG) denoising systems continuously produce a huge volume of signals. IoT sensor devices produce ECG signals at a very rapid rate. As a result, the IoT-based health monitoring system generates ECG signals with very high noise levels. A clean ECG signal is needed for effective heart disease management. Imbalanced electrolytes cause an abnormal ECG reading. The noise can also cause fluctuations the ECG signals. This study shows a novel IoT-based ECG denoising method by combining two filters: the Median Modified Weiner (MMW) and the Extended Kalman filter (EKF), to overcome this issue. The characteristic of ECG signals are first subjected to the MMW filter. The extracted ECG signal is then explained with the Extended Kalman filter. MAT LAB simulates the proposed method. Root mean square error (RMSE), contrast-to-noise ratio (CNR), signal contrast, and coefficient of variation (COV) are used in the proposed MMW-EKF framework to the current systems are compared to Signal-to-noise ratio (SNR). We demonstrate how the suggested technique effectively distinguishes between various ECG signals from a noisy sample input.

***Keywords** – Electrocardiogram; Internet of things; Denoising; Median Modified Weiner Filter; Extended Kalman Filter.*

1. INTRODUCTION

Patients can receive immediate treatment and return to normal activities by using remote ECG monitoring [1,2]. The ability to detect cardiac events earlier results in fewer unnecessary hospitalizations because the patient can be treated at home [3]. In the Intensive Care Unit (ICU) and the operating room, ECG monitoring is now considered standard of care [4]. It is used to identify arrhythmias and ischemia, as well as to evaluate the performance of a pacemaker/automatic implanted cardioverter-defibrillator [5,6]. In this era, it is possible to monitor the vital processes of humans using the IoT technology, regardless of where they are or what they are doing [7, 8]. Patients can monitor and control their health parameters due to the availability and advancement of IoT devices. The combined system offers the user several benefits, including detecting cardiac illness through symptoms in an emergency, transmitting messages to doctors and assisting in its treatment [9-11].

In today's world, everyone has a hectic schedule and leads a fast-paced lifestyle, which increases the risk of heart disease [12]. Cardiovascular disorders are currently the primary cause of death worldwide, emphasizing the importance of using an excellent methodology to assess a patient's cardiac health [13]. One of the most important and widely used medical instruments for heart analysis and disease diagnosis is the ECG [14]. It is a non-invasive treatment used in hospitals for measuring and diagnosing irregular cardiac rhythms. While the ECG signal is being recorded, some noise distortions can interfere [15]. Baseline drift (also known as lower frequency noise), muscle noise, electromagnetic interference and power line interference from various other devices can all cause noise in ECG readings [16,17]. As a result, one of most significant challenges in biomedicine signal processing is the extraction for pure cardio-logical indices from noisy observations, which necessitates dependable strategies to preserve diagnostic data of recorded signal [18,19].

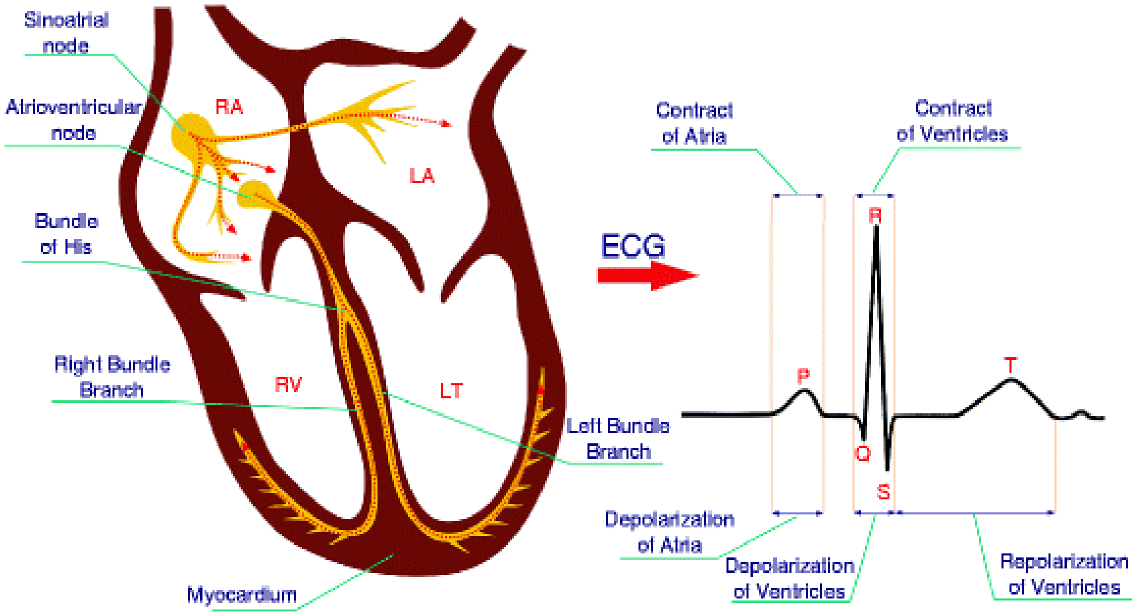


Figure 1. Typical ECG Signal [20, 21]

ECG signals have a frequency range of 0.5 to 100 Hz, and artifacts play a significant part in their processing [22]. It is made up of multiple parts, including the P wave, Q, R, and S waves (QRS) complex. Figure 1 depicts a typical ECG signal, with QRS complex representing high frequency components and T and P representing low frequency components, and any deviation in these parameters indicates the presence of heart irregularities. These waves are caused by ventricular repolarization, ventricular depolarization, and atrial depolarization in the distant field. Conventional filters are frequently employed to remove various undesirable frequency components from ECG data [23-25].

There have been several studies in recent years on ways to reduce these noises prior to disease identification and classification [26]. To remove such noises, a variety of pre-processing approaches are available. Adaptive filtering is one of the strategies for noise reduction in ECG signals, although it consumes a lot of time [27, 28]. The MMF is another common nonlinear filtering strategy for extracting tiny cardiac components from noisy ECGs, although it fails to remove Gaussian noise [29, 30]. The Wavelet Transform (WT) approach for denoising biological signals with multi-resolution characteristics, such as ECG, has gotten much interest [31]. EKF uses a linear model of the expected state with noise-corrupted observations to find the unknown state of a dynamical system. Since most systems were nonlinear, the EKF has been enhanced [32, 33]. Because of this, IoT devices' massive amounts of sensor data are not being stored using standard data processing tools and methodologies [34, 35]. Several techniques for extracting ECG components contaminated by background noise have been proposed [36]. Furthermore, the state model or measurements are unreliable in highly contaminated ECGs. These constraints push us to seek better solutions. The IoT model-based MMW-EKF algorithm has been proposed in this paper as a wiener filtering framework. Because of its nonlinear framework, the proposed algorithm outperformed other non-Gaussian non-stationary algorithms at all input SNRs. As a result, the proposed technique can completely trace the ECG signal even during periods with a lot of noise. In other words, to increase ECG denoising performance while reducing computing complexity by applying the MMW-EKF architecture. However, overall filtering performance is expected to be improved because MMW-EKF has proven to be accurate and efficient in noise removal.

This is how the rest of the study is structured. The reviews that were consulted are included in Section 2. Sections 3 present the proposed methods. Experimental and analytical results are discussed in Section 4. Section 5 discusses the conclusions of the method that has been described.

1.1. Background And Motivation

EKF is an efficient optimal estimator that provides a recursive computational technique for forecasting the state of a discrete-information controlled process from typically noisy observations while simultaneously estimating the estimates' uncertainty. It is used to decrease noise for power line interference. It is used to lessen baseline wander noise. We thoroughly reviewed ECG analysis and presented it as a stages-based process model in order to more precisely characterise and categorise the flow and importance of each phase of ECG signal processing. Considering the significant effects that effective ECG signal analysis has on both public health and the economy, we also conducted this study to provide a perspective on software and hardware instruments, as well as real-time monitoring using portable and wearable devices.

2. LITERATURE REVIEW

In 2021 Lutin et al., [37] created a learning base (QIE) was constructed and 23 photoplethysmography (PPG) datasets from the TROIKA database were studied. To the best of our knowledge, in order to improve heart rate (HR) estimate, the suggested quality engine is first tested using wrist PPG data obtained during various physical activities. When employed in combination with the cutting-edge

Wiener filtering & Phase vocoder (WFPV) method, the QIE improved HR estimate by 43% on average.

In 2021 Cheng et al., [38] proposed a novel approach for automated ECG recognition and categorization. It is used to lessen baseline wander noise. We thoroughly reviewed ECG analysis and presented it as a stages-based process model in order to more precisely characterize and categories the flow and importance of each phase of ECG signal processing. Considering the significant effects that effective ECG signal analysis has on both public health and the economy, we also conducted this study to provide a perspective on software and hardware instruments, as well as real-time monitoring using portable and wearable devices.

In 2021 Salehi and Vahidi, [39] suggested three phases and three denoising filters. I recommend three steps, three denoising filters. The Coefficient of Friction (COF) is calculated from the noise picture in the first phase. The coefficients of variation are then subjected to fuzzy c-means (FCM). The use of FCM results in the fuzzy categorization of picture areas. The three denoising filters are combined in the second stage. Fuzzy logic techniques are used in the third stage to analyze the final image. The experimental results indicate that the proposed denoising method can maintain image featurs and edges when compared to earlier despeckling techniques.

In 2022 Sarafan, et al., [40] proposed a suggested a novel method for extracting the ECG non-invasively from single-channel ECG signals using the ensemble Kalman filter (EnKF) Using the PhysioNet 2013 Challenge bank, the suggested method produces an F1 score of 97.25%, a sensitivity of 96.91%, and a minimum positive predictive value of 97.59. Our findings further demonstrate the effectiveness and dependability of the suggested strategy, which works better than earlier EKF-based algorithms.

In 2022 Tahir, et al., [41] proposed an adaptive noise cancellation (ANC) based on EKF that takes the PLI frequency into account as a different model parameter. As a result, it can follow power line interference (PLI) with a floating frequency. In recursive least squares (SSRLS) state space, filter-based PLI suppression differs from the suggested suppressor's performance. The EKF-based ANC system that is offered performs better than SSRLS-based model, according to the simulation findings. PLI is successfully removed from the ECG in each of the four cases that were examined using the RLS-based ANC approach.

In 2022 Sarafan, et al., [42] proposed the development of a new approach for denoising ECG data based on EnKF. Additional filter techniques that we analyze are the Savitzky-Golay (SG) filter, the ensemble empirical mode decomposition (EEMD), the recursive least squares (RLS) filter, the normalized least mean squares (NLMS) filter, and the total variation denoising technique. To execute. (TVD), wavelet, EKF. A noise stress test database from MIT-BIH where used. Upgraded MIT-BIH database with motion artefacts produces an average SNR of 10.96, a PRD of 150.45, and a correlation value of 0.959 using the recommended approach.

In 2023 Minh, et al., [43]. suggested a cutting-edge framework that uses data mining techniques to extract information related to diagnosing cardiac illness at the network edge while maintaining the integrity of ECG data by eliminating noise. Empirical studies demonstrate that the suggested framework increases real-time detection accuracy while preserving information integrity, in comparison to earlier approaches.

In 2023 Priyadarshini, et al., [44] proposed in order to ensure optimal energy utilization with little battery capacity, we have developed a lightweight solution that prioritizes slope amplification with real-time segmentation methods for complex QRS assessment of pulse rate. The power is provided by local computing on the node. With an enhanced frequency of 269 MHz and a power consumption of 0.7 MW, the Spartan6 FPGA on which the design was based proved perfect for real-time ECG monitoring devices.

In 2023 Cañón-Clavijo, et al., [45] presented an IoT method for monitoring ECG signals and analysing cardiac data to produce an alarm when an arrhythmia is detected. The best classification accuracy for the studied arrhythmias, according to the data, is achieved by the k-nearest neighbor method (94% for premature ventricular beats, 81% for fusion of ventricular beats, and 82% for extra sexual beats). Additionally, it can discriminate 93% and 97%, respectively, between regular and undefinable hits.

These techniques are better than earlier approaches, but they have certain shortcomings as well. Here are a few instances of aberrant ECG readings. These include: Right and left bundle branch block symptoms, premature atrial and ventricular contractions, nonspecific T-wave alterations, and ventricular hypertrophy. To overcome these drawbacks, Efficient MMW and EKF Based ECG Denoising techniques has been proposed.

3. PROPOSED MMW-EKF ALGORITHM

In this paper proposed a novel Efficient MMW-EKF Based ECG Denoising Method has been proposed. The MMWF algorithm is employed to preprocess the noisy ECG signal. As stated, the MMWF estimation uses the signal's characteristic to denoise it. The partially denoised ECG signal is subsequently processed using the EKF approach to solve that problem. To denoise the ECG dynamic signal obtained after MMWF, the EKF discretized it. Finally, the ECG signal is reconstructed by combining all of the EKFs. As a result, the noises' harmful effects were significantly reduced. In Figure 2, a block diagram depicts the proposed method, which successfully combines MMW-EKF methodologies.

This section describes our proposed ECG denoising method in detail. First, the MMW filters is applied to the characteristic ECG signal. The EKF is then utilized to describe the method for extracting the ECG signal from noise.

This approach can maintain the edge signals while the noise distribution is reduced. The following is how the MMWF algorithm is calculated:

$$h_{mmwf}\;(x,y) = \hat{\delta} + \frac{\sigma^2 - u^2}{\sigma^2}\,[a(x,y) - \hat{\delta}] \qquad (1)$$

where $\hat{\delta}$ is the median value's size, $\sigma^2$and $u^2$are the signal's Gaussian noise variance and the noise variation, respectively, and *a (x, y)* is the amplitude at the time (x, y).

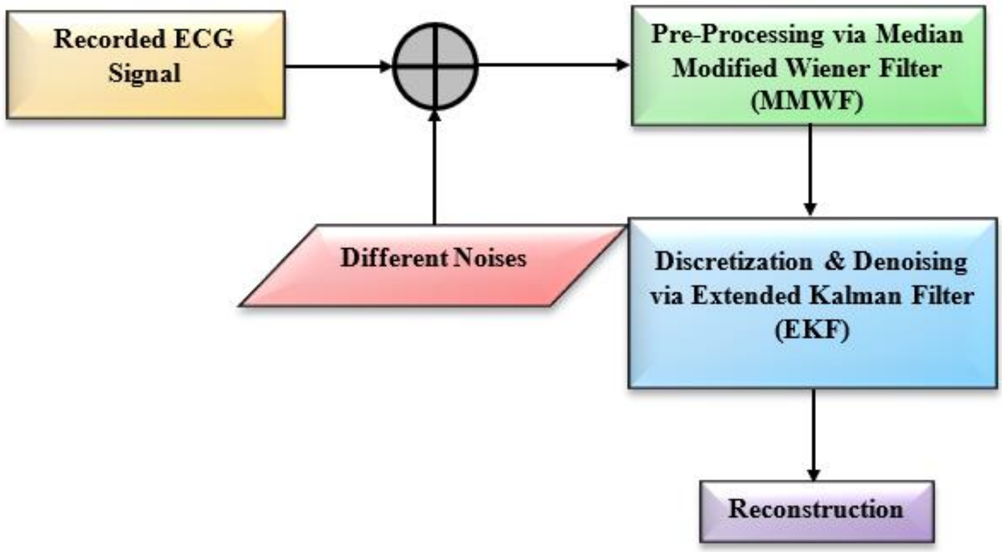

Recorded ECG Signal
Pre-Processing via Median Modified Wiener Filter (MMWF)
Different Noises
Discretization & Denoising via Extended Kalman Filter (EKF)
Reconstruction

Figure 2. Denoising of ECGs: Overall Proposed Architecture

3.1. MMWF Algorithm: ECG Signal Characteristics

The MMWF approach is employed to lessen the distribution of noise in ECG signal. The background area of a deteriorated signal is denoised using the median filter to enhance signal quality. Additionally, the Wiener filter (WF) is generally employed in this method to maintain the edge signal. The MMWF method, which is based on the WF, decreases noise in the deteriorated signal by replacing the values of the mask matrix with the median values. The WF is represented as follows in the r ×t sized mask matrix:

$$\mu = \frac{1}{rt}\sum\nolimits_{r,t\in a} n(r,t)$$

$$\sigma^2 = \frac{1}{rt}\sum\nolimits_{r,t\in a} n^2(r,t) - \mu^2$$

Where, µ is the mean, σ^2 is the variance of Gaussian noise in the signal, r × t is the size of the neighborhood area an in the mask, and n (r, t) represents each value in the area a.

Thus, the MMWF is represented as follows:

$$k_{mmwf}(r,t) = \mu + \frac{\sigma^2 - v^2}{\sigma^2}(b(r,t)) - \mu)$$

where v^2 is the mask WF matrix's noise variance setting.

The MMWF technique has the benefit of enhancing the poor signal quality in the following ways: Comparing the drop-off effect to the median and Wiener filter approaches, the edge signal is better retained. In conclusion, the MMWF approach performs better than traditional filters in terms of denoising effect and can keep the edge information while removing the background noise signal.

3.1.1. Evaluation of Quality

ECG signals are analyzed by MMWF, and reconstructed signals were acquired. The CNR, COV, and SNR linked to noise are analyzed.

The degree of dispersion around the mean is inversely correlated with the COV. It is frequently stated as a percentage. [46,47] It enables for the comparison of value distributions with disparate measurement scales because it lacks units. The strength of the signal is contrasted with the strength of the noise in a SNR. Decibels are most frequently used to express it (dB). Higher numbers often denote a better specification since there is a greater ratio of valuable information (the signal) to undesired data (the noise). CNR is similar to the metric SNR.

Because COV and CNR are derived using a single region's signal and variance values, they were employed for noise level analysis. Furthermore, the CNR was used to calculate the signal difference between two neighboring regions' variance values while also taking noise and contrast into account.

$$COV = \frac{\sigma_X}{\delta} \qquad (2)$$

$$SNR = \frac{S_X}{\sigma_X} \qquad (3)$$

$$CNR = \frac{|S_X - S_Y|}{\sqrt{\sigma^2{}_X + \sigma^2{}_Y}} \qquad (4)$$

The Return on Investment (ROI's) average standard deviation and signal intensity are S_X and σ_X, respectively, whereas the background region's average standard deviation and signal intensity are S_X and σ_X. The ROI is used to mould the image compression to one particular area of compression. Instead of reducing all the pixel intensities we are using ROI to specify the required arrangements.

In addition, for the purpose of determining the loss of high-frequency signals caused by MMWF technique, signal similarity was evaluated with reference to noise-free signals. Applied the following parameters for this evaluation: RMSE, cubic centimetre (CC) and Peak Signal Noise Ratio (PSNR):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(r_i - c_i)^2}{N}} \quad (5)$$

$$PSNR = 10(\frac{{S_{max}}^2}{RMSE^2}) \quad (6)$$

The reference and comparison signals are represented by r_i and c_i, respectively, and ${S_{max}}^2$ is maximum signal intensity in ROI.

$$CC = \frac{\sum_{i=1}^{N}(r_i - \hat{r})(c_i - \hat{c})}{\sqrt{\sum_{i=1}^{N}(r_i - \hat{r})^2}\sqrt{\sum_{i=1}^{N}(c_i - \hat{c})^2}} \quad (7)$$

The $\hat{c}$ and $\hat{r}$ represents the average values of comparison and reference signals.

The signal loss is induced by MMWF smoothing can be assessed using PSNR and RMSE, which quantitatively show the signal difference. COV, CNR and SNR linked to noise was analyzed to evaluate MMWF's noise reduction efficiency from the obtained signals, allowing overall quality of signal to be evaluated. Thus, characteristics of ECG signal are evaluated.

3.2. EKF To Train MMWF for ECG Denoising (MMW-EKF)

The EKF is utilized to denoise the ECG signal once the parameters have been analyzed. Since the nonlinear dynamic ECG model Eqn. (5), (6), and (7) are continuous time, and the EKF is a discrete-time technique, continuous nonlinear dynamic ECG model Eqn. (5), (6), and (7) must be discretized. A first-order numerical method called the Euler methods, often referred to as the forward Euler methods, is used to resolve ordinary differential equations (ODEs) with a specified beginning value. Euler method is used to discrete Eqn. (5), (6), and (7). As a result, the discrete form of (7) will be:

$$a(u+1) = (1+\rho h)a(u) - \omega h b(u)$$

$$b(u+1) = (1+\rho h)b(u) + \omega h a(u)$$

$$c(u+1) = -\sum_{j\epsilon\{P,Q,R,S,T\}} \frac{a_j}{{b_j}^2} h\Delta\theta_j \, exp\, exp\left(-\frac{\Delta\theta_j^2}{2b_j^2}\right) - ((h-1)c(u) - hc_0) \quad (8)$$

where h is the sampling time.

The following is a more compact rewrite of the nonlinear discrete ECG model (26):

$$X_{u+1} = f(X_u) \quad (9)$$

where X_u is the state vector, and it is represented by $X_u = [a_u \; b_u \; c_u]^T$.

$$\begin{pmatrix} a(u+1) \\ b(u+1) \end{pmatrix} = \begin{matrix} (1+\rho h)a(u) - \omega h b(u) \\ (1+\rho h)b(u) + \omega h a(u) \end{matrix}$$

$$c(u+1) - \sum_{j\epsilon\{P,Q,R,S,T\}} \frac{a_j}{{b_j}^2} h\Delta\theta_j \, exp\, exp\left(-\frac{\Delta\theta_j^2}{2b_j^2}\right) - ((h1)c(u) - hc_0) \quad (10)$$

The state equation of the discrete ECG model without noise is represented by the vector equation (28). We need to introduce some random sounds in (27) to simulate a more realistic ECG signal:

$$X_{u+1} = f(X_u, r_u) \quad (11)$$

where $r_u = [r_1, r_2, r_3]^T$state noise is an additive random vector that is normal and Gaussian with a zero mean, then (28) can be written as:

$$\begin{pmatrix} a(u+1) \\ b(u+1) \\ c(u+1) \end{pmatrix} = \begin{pmatrix} (1+\rho h)a(u) - \omega h b(u) + r_1(u) \\ (1+\rho h)b(u) + \omega h a(u) + r_2(u) \\ -\sum_{j\epsilon\{P,Q,R,S,T\}} \frac{a_j}{{b_j}^2} h\Delta\theta_j exp exp\left(-\frac{\Delta\theta_j^2}{2b_j^2}\right) - ((h-1)c(u) - hc_0) + r_1(u) \end{pmatrix} \quad (12)$$

The state vector $X_u = [a_u \; b_u \; c_u]^T$can be connected to the measurement equation corresponding to state space representation (29) by the following relation:

$$b_u = g\,(X_u, m_u)$$

$$= [0\;0\;1]\; X_u + m_u \quad (13)$$

m_u is measurement noise and y_u is the considered measure.

A linear approximation of (30) is required to use EKF (see (2) and (3)). To compute the Jacobian matrices entries (see (3)), arrange (30) and (31) as follows:

$$\begin{cases} a(u+1) \quad X\,(\,a(u), b(u), c(u), r_1(u)) \\ b(u+1) = Y\,(\,a(u), b(u), c(u), r_2(u)) \\ c(u+1) \quad Z\,(a(u), b(u), c(u), r_3(u)) \end{cases} \quad (14)$$

$$b_u = g\,(X_u, m_u) \quad (15)$$

The entries of the Jacobian matrix A_k are then calculated.

$$\frac{\partial X}{\partial a} = 1 + h - \left(\frac{2ha(u)^2 + hb(u)^2}{\sqrt{a(u)^2 + b\,(u)^2}}\right)$$

$$\frac{\partial X}{\partial b} = -\frac{ha(u)b(u)}{\sqrt{a(u)^2 + b(u)^2}} - \omega h \frac{\partial X}{\partial c} = 0$$

$$\frac{\partial Y}{\partial a} = -\frac{ha(u)b(u)}{\sqrt{a(u)^2 + b(u)^2}} + \omega h$$

$$\frac{\partial Y}{\partial b} = 1 + h - \left(\frac{ha(u)^2 + 2hb(u)^2}{\sqrt{a(u)^2 + b(u)^2}}\right)\frac{\partial Y}{\partial c} = 0$$

$$\frac{\partial Z}{\partial a} = \sum_{j\epsilon\{P,Q,R,S,T\}} \frac{\frac{a_{j\omega}}{{b_j}^2} hy(u)}{a(u)^2 + b(u)^2} exp\, exp\left(-\frac{\Delta\theta_j^2}{2b_j^2}\right) - [\,1 - \frac{\Delta\theta_j^2}{b_j^2}] \quad (16)$$

$$\frac{\partial Z}{\partial y} = \sum_{j\epsilon\{P,Q,R,S,T\}} \frac{-\frac{a_{j\omega}}{{b_j}^2} hx(u)}{x(u)^2 + y(u)^2} exp\, exp\left(-\frac{\Delta\theta_j^2}{2b_j^2}\right) - [\,1 - \frac{\Delta\theta_j^2}{b_j^2}]$$

$$\frac{\partial Z}{\partial z} = 1 - h$$

$$\frac{\partial X}{\partial r_1} = \frac{\partial Y}{\partial r_2} = \frac{\partial Z}{\partial r_3} = 1$$

$$\frac{\partial X}{\partial r_2} = \frac{\partial X}{\partial r_3} = \frac{\partial Y}{\partial r_1} = \frac{\partial Y}{\partial r_3} = \frac{\partial Z}{\partial r_1} = \frac{\partial Z}{\partial r_2} = 0$$

$$\frac{\partial g}{\partial x} = \frac{\partial g}{\partial y} = 0; \frac{\partial g}{\partial z} = 1 - h; \frac{\partial g}{\partial m} = 1$$

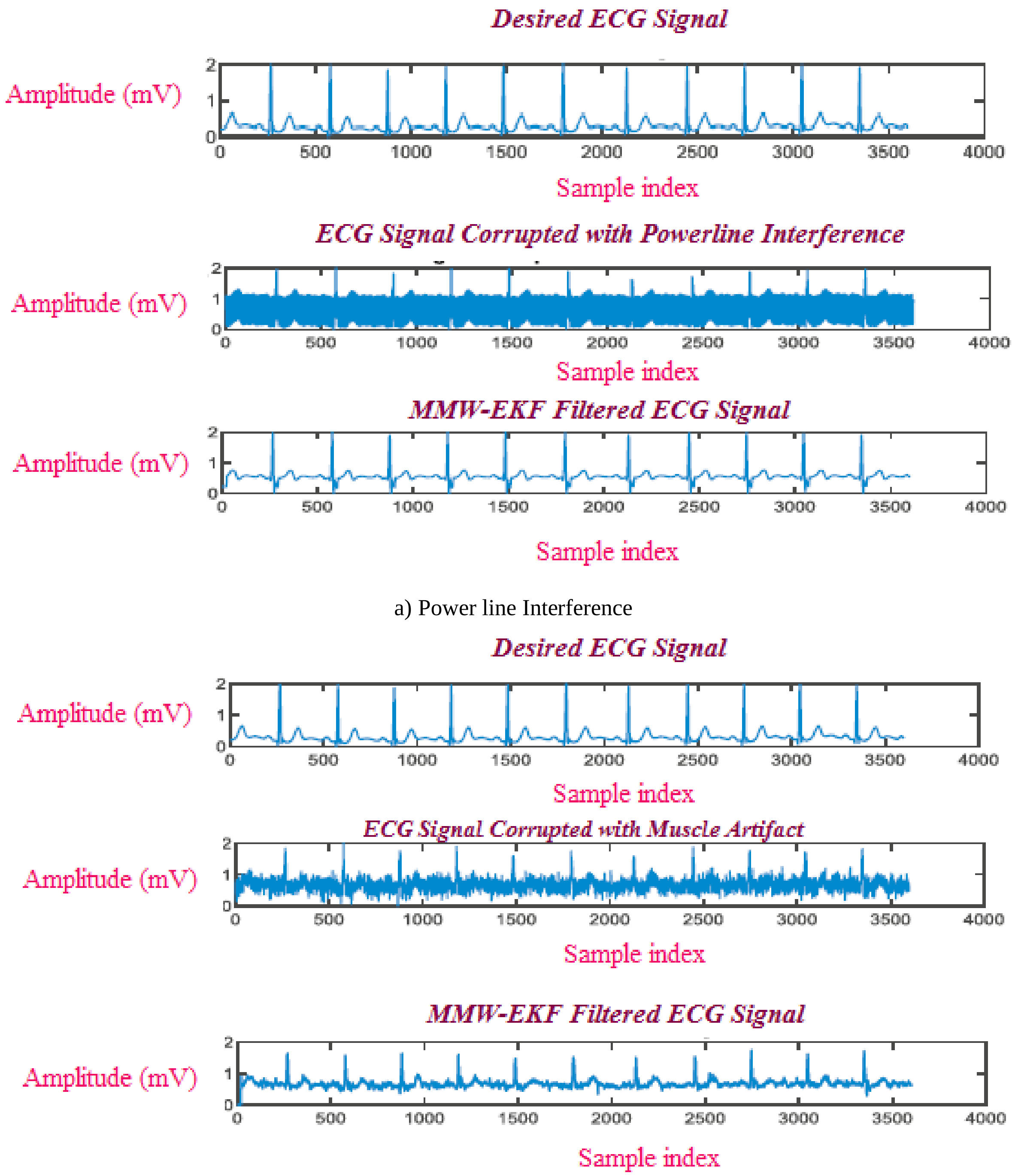
For noise removal, the proposal utilizes the (Enterprise data management) EDM implementation. However, our method constructs an MMW-EKF filter using the dynamical set of equations, which uses the state dynamics and ECG as an observation. As a result, the proposed MMW-EKF technique for ECG denoising is an efficient.

4. RESULT AND DISCUSSION

The experimental arrangement of the proposed MMM-EKF technique based denoised was implemented using MATLAB. A comparison of the proposed MMM-EKF technique Performance with some filters they are Gaussian Filter (GF), Median Filter (MF) and CF is made. The proposed MMW-EKF based on ECG denoising is simulated in this section using MATLAB. Figure 3 depicts the filter outputs.

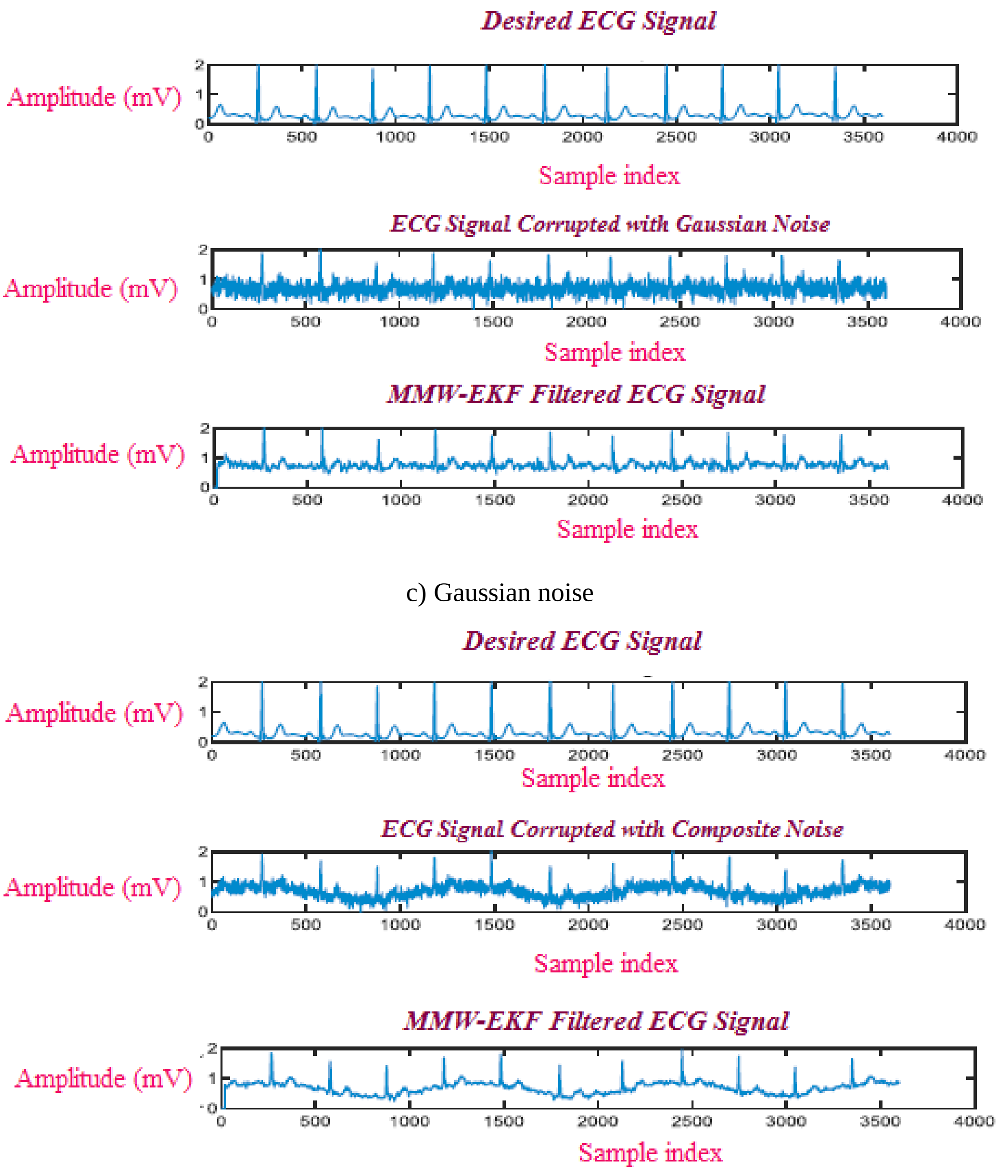
Datasets

A distinct clinical ECG dataset that has been optimized for machine learning computers is the PTB-XL database. 21,837 clinical 12-lead ECGs, each lasting 10 seconds and recorded at 500 Hz and 100 Hz with 16-bit resolution, are included in the PTB-XL ECG dataset. These ECGs are from 18,885 different people.



a) Power line Interference

b) Muscle artefact



c) Gaussian noise

d) Composite noise

Figure 3. Graphical Illustration for a Denoised ECG Signal using the Median Modified Wiener and Extended Kalman Filters with Various Noises

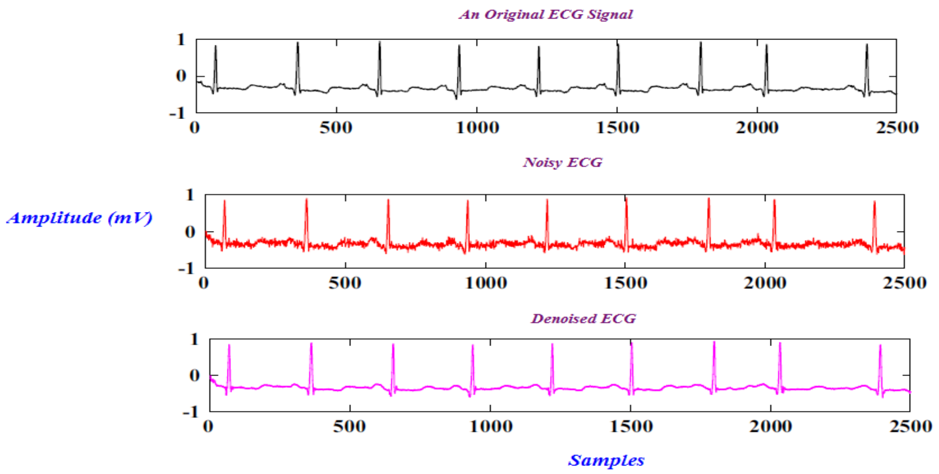


Figure 4. Proposed Method for a Denoised Signal

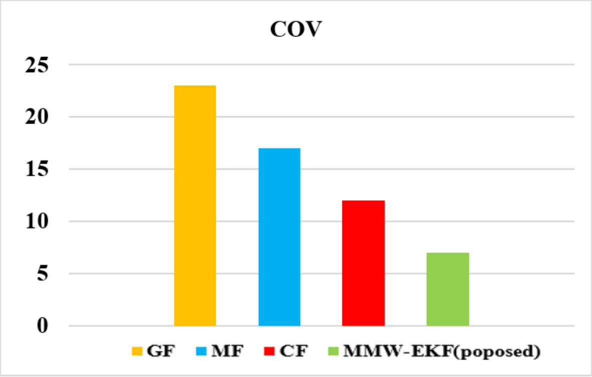
Figure 4 displays the original, clean ECG signal, the noisy variations in the signal, and the denoised signal produced by the suggested technique. With a 10dB input SNR, the graphs exhibited database record number 100 in the MMW-EKF.

When comparing the Kalman, Wiener, and proposed MMW-EKF filters to various data sets, SNR and RMSE were determined for each data set. Our proposed approach has a greater SNR and a lower RMSE (see Table 1).

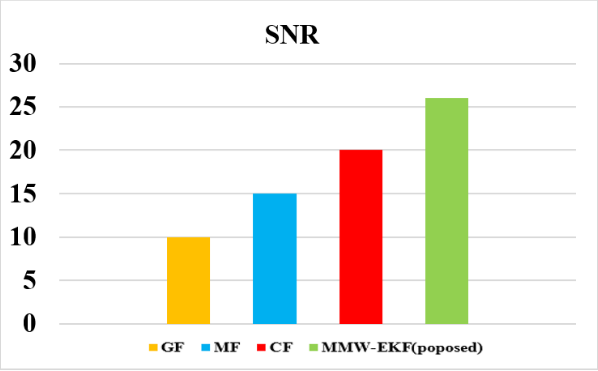
Table 1. Different noise has different values for various performance parameter

SIGNAL EXPECTATION									
Noises	**Filters**	**SNR**				**RMSE**			
		Data 103	**Data 105**	**Data 121**	**Average**	**Data 103**	**Data 105**	**Data 121**	**Average**
PI	WF	6.985	6.435	9.245	7.565	0.062	0.132	0.051	0.072
	KF	4.773	5.435	6.422	5.546	0.165	0.156	0.115	0.143
	MMW-EKF	7.123	6.945	9.545	7.856	0.051	0.111	0.021	0.0371
Gaussian	WF	5.352	5.986	8.216	6.593	0.122	0.148	0.073	0.121
	KF	4.455	5.255	6.768	5.458	0.221	0.186	0.144	0.184
	MMW-EKF	5.455	6.203	8.521	6.956	0.101	0.122	0.055	0.026
Muscle Artifact	WF	6.846	5.344	8.979	7.048	0.073	0.185	0.056	0.102
	KF	5.321	5.236	7.426	6.012	0.131	0.162	0.093	0.142
	MMW-EKF	7.235	5.682	9.235	7.3315	0.052	0.098	0.045	0.065
Composite	WF	6.435	4.911	7.347	6.247	0.063	0.234	0.095	0.142
	KF	4.726	5.188	6.325	5.349	0.160	0.163	0.152	0.159
	MMW-EKF	6.985	5.356	7.563	6.5641	0.044	0.132	0.054	0.084

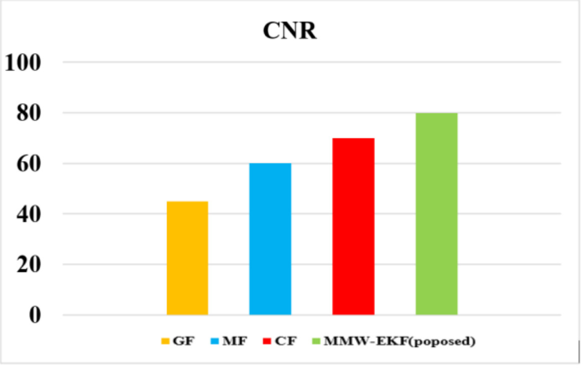
The CNR, COV and SNR, were calculated to compare the noise levels of the MMW-EKF applied signal to the original signal and other existing approaches. Over the original signal and several existing approaches, the CNR, COV and SNR values of the MMW-EKF reconstructed signal are increased by factors of 2.01, 1.02, and 1.60, respectively (see Figure 5).



a) COV



b) SNR



c) CNR

Figure 5. Comparison between the Original Signal and the Signal Reconstructed with MMW-EKF

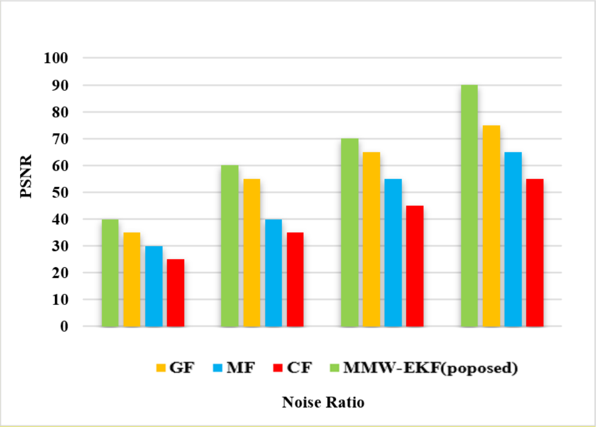


Figure 6. Measures of PSNR

Our results indicate that the MMW-EKF can successfully reduce noise from ECG signals. Furthermore, the sample should be optimized for signal loss and noise level to improve MMW-EKF efficiency. Measures of PSNR is shown in Figure 6.

The PSNR between two images - the actual image and its noisy approximation - is used to describe noise. The maximum pixel value for 8-bit pictures is 255. PSNR values for image compression are often in the 30 to 50 dB range.

5. CONCLUSION

The MMW-EKF were used in this paper to design an IoT based ECG denoising process that allowed advantage of the denoising features of each filter. The proposed technique denoises a noisy signal by preprocessing it and estimated the signal's characteristics with the MMW filter. To further decrease the effects of additive noise, the partially denoised ECG signal is analyzed and discretized using an EKF. The performance of parameters such as COV, SNR, and CNR is also analyzed in the simulation results. The analysis of this research confirms the proposed method's suitable for filtering noisy ECG signals. In the event of a sudden illness, the terminal will also have the ability to collect data for processing and notify the patient's family of a data transfer. To reduce distortion and boost system reliability, additional work entails adding baseline variation to the EDM.

CONFLICTS OF INTEREST

The writers say they have no competing interests.

FUNDING STATEMENT

Not applicable.

ACKNOWLEDGEMENTS

The author would like to express his heartfelt gratitude to the supervisor for his guidance and unwavering support during this research for his guidance and support.

REFERENCES

[1] G.J. Lakshmi, M. Ghonge and A.J. Obaid, "Cloud based Iot smart healthcare system for remote patient monitoring", *EAI Endorsed Transactions on Pervasive Health and Technology*, vol. 7, no. 28, pp. e 4-e4, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[2] Y. Feng and Z. Pan, "Optimization of remote public medical emergency management system with low delay based on internet of things", *Journal of Healthcare Engineering*, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[3] G. Boriani, F. Guerra, R. De Ponti, A. D'Onofrio, M. Accogli, M. Bertini, G. Bisignani, G.B. Forleo, M. Landolina, C. Lavalle and P. Notarstefano, "Five waves of COVID-19 pandemic in Italy: Results of a national survey evaluating the impact on activities related to arrhythmias, pacing, and electrophysiology promoted by AIAC (Italian Association of Arrhythmology and Cardiac Pacing)", *Internal and Emergency Medicine*, vol. 18, no. 1, pp. 137-149, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[4] M.C. Perez-Guzman, E. Duggan, S. Gibanica, S. Cardona, A. Corujo-Rodriguez, A. Faloye, M. Halkos, G.E. Umpierrez, L. Peng, G.M. Davis and F.J. Pasquel, "Continuous glucose monitoring in the operating room and cardiac intensive care unit". *Diabetes Care*, vol. 44, no. 3, pp. e50-e52, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[5] C. Li, X. Hu and L. Zhang, "The IoT-based heart disease monitoring system for pervasive healthcare service", *Procedia computer science*, vol. 112, pp. 2328-2334, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[6] M. Bianco, S. Breviario, T. Fraccalini, R. Ferri, C. Biolè, P. Destefains, G. Varalda, A. Boccuzzi and A. Chinaglia, "Epilepsy and syncope-A case report and narrative review of arrhythmias connected to temporal lobe epilepsy", *Journal of Electrocardiology*, vol. 73, pp.76-78, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[7] A.K.M. Majumder, Y.A. ElSaadany, R. Young and D.R. Ucci. "An energy efficient wearable smart IoT system to predict cardiac arrest", *Advances in Human-Computer Interaction*, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[8] Naji, Hassan Saadallah, and Hanan Badeea Ahmed, "Design and Analysis of a Signal Denoising and Estimation System", *In IOP Conference Series: Materials Science and Engineering*, vol. 1067, no. 1, pp. 012145. IOP Publishing, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[9] Hesar, Hamed Danandeh, and Maryam Mohebbi. "An adaptive particle weighting strategy for ECG denoising using marginalized particle extended Kalman filter: An evaluation in arrhythmia contexts", *IEEE journal of biomedical and health informatics*, vol. 21, no. 6, pp. 1581-1592, 2017.[CrossRef] [Google Scholar] [Publisher Link]

[10] K. Azbeg, O. Ouchetto and S.J. Andaloussi, "BlockMedCare: A healthcare system based on IoT, Blockchain and IPFS for data management security", *Egyptian Informatics Journal*, vol. 23, no. 2, pp.329-343, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[11] N.Y. Philip, J.J. Rodrigues, H. Wang, S.J. Fong and J. Chen. "Internet of Things for in-home health monitoring systems: Current advances, challenges and future directions", *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 2, pp. 300-310, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[12] B. R. Manju and M. R. Sneha. "ECG denoising using wiener filter and kalman filter." *Procedia Computer Science*, vol. 171, pp. 273-281, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[13] S. Gaamouri, M. Bousbia Salah, and R. Hamdi. "Denoising ECG signals by using extended kalman filter to train multi-layer perceptron neural network", *Automatic Control and Computer Sciences*, vol. 52, no. 6, pp. 528-538, 2018[CrossRef] [Google Scholar] [Publisher Link]

[14] McSharry, E. Patrick Gari D. Clifford, Lionel Tarassenko, and Leonard A. Smith, "A dynamical model for generating synthetic electrocardiogram signals", *IEEE transactions on biomedical engineering*, vol. 50, no. 3, pp. 289-294, 2003. [CrossRef] [Google Scholar] [Publisher Link]

[15] Lahmiri, Salim, and Mounir Boukadoum, "Hybrid Wiener and partial differential equations filter for biomedical image denoising", *In 2016 14th IEEE International New Circuits and Systems Conference (NEWCAS)*, IEEE, pp. 1-4, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[16] A. Turnip, M. Turnip, A. Dharma, D. Paninsari, T. Nababan and C.N. Ginting, "An application of modified filter algorithm fetal electrocardiogram signals with various subjects", *International Journal of Artificial Intelligence*, vol. 18, no. 1, pp.207-217, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[17] Kaur, Harjeet, and Rajni. "A novel approach for denoising electrocardiogram signal using hybrid technique", *Journal of Engineering Science and Technology*, vol. 12, no. 7, pp. 1780-1791, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[18] R.S. Chandel, S. Sharma, Kaur, S., Singh, S. and R. Kumar, "Smart watches: A review of evolution in bio-medical sector", *Materials Today: Proceedings*, vol. 50, pp.1053-1066, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[19] Ouali, Mohammed Assam, Kheireddine Chafaa, Mouna Ghanai, Louis Moreno Lorente, and Dolores Blanco Rojas. “ECG denoising using extended Kalman filter*”, In 2013 International Conference on Computer Applications Technology (ICCAT)*, pp. 1-6. IEEE, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[20] J. Duan, Q. Wang, B. Zhang, C. Liu, C. Li and L. Wang, “Accurate detection of atrial fibrillation events with RR intervals from ECG signals”, *Plos one*, vol. 17, no. 8, pp. e0271596, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[21] A. Ukil, L. Marin, S.C. Mukhopadhyay and A.J. Jara. “AFSense-ECG: Atrial fibrillation condition sensing from single lead electrocardiogram (ECG) signals”, *IEEE Sensors Journal*, vol. 22, no. 12, pp. 12269-12277, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[22] N. Keskes, S. Fakhfakh, O. Kanoun and N. Derbel. “Representativeness consideration in the selection of classification algorithms for the ECG signal quality assessment”, *Biomedical Signal Processing and Control*, vol. 76, p.103686, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[23] A. Mishra, and S. Bhusnur, “A New Adaptive Modeling and Denoising of Real ECG signal”, *In 2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT)* IEEE, pp. 1-6, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[24] D.U. Uguz, Z.T. Canbaz, C.H. Antink, M. Lüken and S. Leonhardt, “A novel sensor design for amplitude modulated measurement of capacitive ECG”, *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1-10, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[25] V.A. Ardeti, V.R. Kolluru, G.T. Varghese and R.K. Patjoshi. “An Overview on State-of-the-Art Electrocardiogram Signal Processing Methods: Traditional to AI-Based Approaches”, *Expert Systems with Applications*, pp. 119561, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[26] Yazdanpanah, Babak, G. S. N. Raju, and K. Sravan Kumar, “Reduction Noise of ECG Signal Using Extended Kalman Filter”, *International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)*, vol. 3, no. 9, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[27] D. Liu, H. Zhao, X. He and L. Zhou, “Polynomial constraint generalized maximum correntropy normalized subband adaptive filter algorithm”, *Circuits, Systems, and Signal Processing*, pp.1-18, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[28] Khazraj, Hesam, F. Faria Da Silva and Claus Leth Bak, “A performance comparison between extended Kalman Filter and unscented Kalman Filter in power system dynamic state estimation”, *In 2016 51st International Universities Power Engineering Conference (UPEC)*, pp. 1-6. IEEE, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[29] V. Ramakrishnan and P. Pete, “Weighted Median filtering-based fusion of multiple exposure images for HDRI. D. J, Weighted Median filtering-based fusion of multiple exposure images for HDRI”, April 9, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[30] Bakhshande, Fateme, and Dirk Söffker, “Adaptive Step Size Control of Extended/Unscented Kalman Filter Using Event Handling Concept”, *Frontiers in Mechanical Engineering*, vol. 5, no. 74, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[31] Kulikov, Gennady Yu, and Maria V. Kulikova, “Practical implementation of extended Kalman filtering in chemical systems with sparse measurements”, *Russian Journal of Numerical Analysis and Mathematical Modelling*, vol. 33, no. 1, pp. 41-53, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[32] Zhao, Junbo, Marcos Netto and Lamine Mili, “A robust iterated extended Kalman filter for power system dynamic state estimation”, *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 3205-3216, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[33] Becerra, Victor Manuel, P. D. Roberts and G. W. Griffiths. “Applying the extended Kalman filter to systems described by nonlinear differential-algebraic equations”, *Control Engineering Practice*, vol. 9, no. 3, pp. 267-281, 2001. [CrossRef] [Google Scholar] [Publisher Link]

[34] Gustafsson, Fredrik, and Gustaf Hendeby, “Some relations between extended and unscented Kalman filters”, *IEEE Transactions on Signal Processing*, vol. 60, no. 2, pp. 545-555, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[35] Y. Zhong, L. Chen, C. Dan and A. Rezaeipanah, “A systematic survey of data mining and big data analysis in internet of things,” *The Journal of Supercomputing*, pp. 1-49, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[36] Shamsollahi, Mohammad Bagher. “ECG denoising and compression using a modified extended Kalman filter structure”, *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 9, pp. 2240-2248, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[37] E. Lutin, D. Biswas, N. Simões-Capela, C. Van Hoof and N. Van Helleputte, “Learning based Quality Indicator Aiding Heart Rate Estimation in Wrist-Worn PPG”, *In 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)* IEEE, pp. 7063-7067, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[38] J. Cheng, Q. Zou and Y. Zhao, “ECG signal classification based on deep CNN and BiLSTM”, *BMC medical informatics and decision making*, vol. 21, pp.1-12, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[39] H. Salehi and J. Vahidi, “A novel hybrid filter for image despeckling based on improved adaptive wiener filter, bilateral filter and wavelet filter”. *International Journal of Image and Graphics*, vol. 21, no. 03, pp. 2150036, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[40] S. Sarafan, T. Le, M.P. Lau, A. Hameed, T. Ghirmai and H. Cao, “Fetal electrocardiogram extraction from the mother’s abdominal signal using the ensemble kalman filter”, *Sensors*, vol. 22, no. 7, p.2788, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[41] S. Tahir, M.M. Raja, N. Razzaq, A. Mirza, W.Z. Khan, S.W. Kim and Y.B. Zikria, “Extended Kalman Filter-Based power line interference canceller for electrocardiogram signal”, *Big Data*, 10(1), pp. 34-53, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[42] S. Sarafan, H. Vuong, D. Jilani, S. Malhotra, M.P. Lau, M. Vishwanath, T. Ghirmai, and H. Cao, “A novel ecg denoising scheme using the ensemble kalman filter”, *In 2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 2005-2008, IEEE, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[43] Q.T. Minh, P.H. Phung and P.N. Huu. “Fog-Enabled IoT Framework for Heart Disease Diagnosis Systems”, *Journal of Mobile Multimedia*, pp. 389-418, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[44] R. Priyadarshini, N. Shaikh, R.K. Godi, P.K. Dhal, R. Sharma and Y. Perwej, “IOT-based power control systems framework for healthcare applications”, *Measurement: Sensors*, vol. 25, pp.100660, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[45] R.E. Cañón-Clavijo, C.E. Montenegro-Marin, P.A. Gaona-Garcia and J. Ortiz-Guzmán. “IoT Based System for Heart Monitoring and Arrhythmia Detection Using Machine Learning”, *Journal of Healthcare Engineering*, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[46] K. Madurakavi and I, J.R., “Assessing nitrogen dioxide (NO2) impact on health pre-and post-COVID-19 pandemic using IoT in India”, *International Journal of Pervasive Computing and Communications*, vol. 18, no. 5, pp. 476-484, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[47] V. Gomathy, K. Janarthanan, F. Al-Turjman, R. Sitharthan, M. Rajesh, K. Vengatesan and T.P. Reshma, “Investigating the spread of coronavirus disease via edge-AI and air pollution correlation”, *ACM Transactions on Internet Technology*, vol. 21, no. 4, pp. 1-10, 2021. [CrossRef] [Google Scholar] [Publisher Link]

AUTHORS

L. Jenifer is a research scholar at Satyabhama institute of science and technology, Chennai, india. She earned a bachelor’s degree in electrical and electronics engineering in 2012 and a master’s degree in VLSI design in 2014 from anna university, Chennai, india. Her research interest includes bio energy, self-sustainable system, waste utilisation, battery monitoring systems, microbial fuel cells and low power electronics.

Xiaochun Cheng Xiaochun Cheng received the Bachelor degree in Computer Engineering in 1992 and PhD in Computer Science in 1996 from Jilin University. He has been EU Project Coordinator at Department of Computer Science in Middlesex University in London since 2012, a guest professor at Jilin University, Beijing Normal University, Northeast Normal University. He published at top rank journals (such as IEEE Transactions on Computers, Information Sciences, Theoretical Computer Science) and international flagship conferences (such as IJCAI, IEEE ICC, IEEE GlobeCom, IEEE SMC). Contributed for peer reviewed 96 published journal papers, peer reviewed 130 conference papers, with five times best conference paper awards by end of 2018. Three papers are in the 2018 top 1% of the academic field of Computer Science based on a highly cited threshold for the field and publication year by Data from Essential Science Indicators.

A. Ahilan received Ph.D. from Anna University, India, and working as an Associate Professor in the Department of Electronics and Communication Engineering at PSN College of Engineering and Technology, India. His area of interest includes FPGA prototyping, Computer vision, the Internet of Things, Cloud Computing in Medical, biometrics, and automation applications. Served Guest editor in several journals of Elsevier, Benthom, IGI publishers. Also, have contributed original research articles in IEEE Transactions, SCI, SCIE, and Scopus indexed peer-review journals. He presented various international conference events like ASQED (Malaysia), ESREF (France). He is doing as a reviewer in IEEE Industrial Informatics, IEEE Access, Measurement, Multimedia Tools & Applications, Computer Networks, Medical systems, Computer & Electrical Engineering, neural computing and applications, Cluster Computing, IET Image Processing, and so on. He has IEEE and ISTE membership. He has worked as a Research Consultant at TCS, Bangalore, where he has guided many computer vision projects and Bluetooth Low Energy projects. Hands on programming in MATLAB, Verilog and python at various technical institutions around India.

P. Josephin Shermila She was born in Kanyakumari District, Tamilnadu, India in 1983. She received her B.E. degree in Electronics and Communication Engineering from Noorul Islam college of Engineering, Kumaracoil, Anna University, India in 2005, and obtained M.E. degree in Computer Communication Engineering from National Engineering College, Kovilpatti, Anna University, India in 2007. She has completed her research in Information and Communication Engineering in Anna University, Chennai, India in 2021. She worked as a Programmer Analyst in Cognizant Technology Solutions from October 2007 to October 2010. She is in teaching profession since November 2010. Currently she is working as Associate Professor in the Department of Artificial Intelligence and Data Science, RMK College of Engineering and Technology, Thiruvallur District- 601206, India. She is a member of few professional bodies and have given few guest lectures in reputed organizations She has published more than 19 articles and has published 15 conference papers. Her research area of interest is Nutrition Estimation from Food Images, Image Processing, Machine Learning and Deep Learning.

Arrived: 10.11.2023
Accepted: 15.12.2023

International Journal of Data Science and Artificial Intelligence (IJDSAI)
Volume 01, Issue 02, November – December (2023)

RESEARCH ARTICLE

DETECTION OF VIOLENCE IN FOOTBALL STADIUM THROUGH BIG DATA FRAMEWORK AND DEEP LEARNING APPROACH

M. Dhipa[1, *] and D. Anitha[2]

[1]Associate Professor, Department of Biomedical Engineering, Nandha Engineering College, Erode 638052 India.
[2]Associate Professor, Department of Information Technology, Muthayammal Engineering College, Namakkal, Tamil Nadu, India.

[*]*Corresponding e-mail: dhipa02m@outlook.com*

Abstract – Football is the most famous game in the world, with over 4 billion supporters worldwide. Football hooliganism refers to the aggressive or destructive actions of a supporter or player in a stadium while watching or participating in a game. To avoid violence, a real-time violence detection system is required to observe the audience and players behaviour in order to take appropriate action before violence occurs. The input of the system is a massive volume of real-time video feeds from various sources, that are processed using the Flink structure. Using the Histogram of Oriented Gradients (HOG) function in the Flink framework, pictures are partitioned into frames and their characteristics are retrieved. The frames are then labelled on the basis of attributes including Groundside-violence model, Crowdside-violence model, human part model, and Non-violence model, are utilised to train the multihead attention based Bidirectional Long Short-Term Memory network for violent scene detection. The RWF-2000 dataset, which contains the training set (80%) and the test set (20%) was used to train the network and also a dataset comprising 410 video footages with non-violence scenes and 409 video footages with violent situations is created by the videos obtained from a football stadium, to make the algorithm more strong to violence detection. Other existing approaches are used to validate the model's performance. When compared with existing systems, the proposed violence detection methodology significantly increases accuracy upto 1.6453%, precision upto 0.646%, recall upto1.959%, and reduces execution time upto 60% than other existing methods.

***Keywords** – Multi-head attention, LSTM, Bidirectional-LSTM, RWF-2000, violence detection, Flink framework*

1. INTRODUCTION

With the expansion and improvements in the domain of computer vision over the last decade, a huge amount of current approaches have arisen and enticed a lot of attention from researchers owing to their wide range of surveillance applications [1]. Video surveillance is crucial in detecting human behaviour and preventing or reducing violence in real time [2]. Violence detection is a critical step to identify regular human activities from abnormal/violent acts in the development of automated security surveillance systems. Regular life interacting actions, such as walking, hand waving, running, and jogging, are frequently classified as normal human activities. Violence, on the other hand, is exposed to unusually ferocious activities, such as fights involving two or more persons [3].

Surveillance using complete human operators has also shown to have several flaws, such as high staffing costs, variability in long-duration capture, and poor multi-screen monitoring capability [4]. The identification and tracking of moving object are the key problems in computer vision-based surveillance systems [9]. Moving object detection is the technique of recognizing a thing in an audiovisual that is shifting its position in relation to the scene's turf of opinion [5]. Hence, an automatic violence detection system is needed [4]. As a result, big data analytics employing deep learning approaches may be able to identify violence more quickly and accurately.

A type of AI and ML that mimics the learning process of humans is deep learning. Deep learning has been more significant in big data analytic solutions in recent years [6]. Big data [8] is an area which involves ways for testing, consistently extracting data from, or deals with data volumes which are overlarge or complicate for traditional data-processing application software to manage. Deep Learning [10] is a strong technique for Big Data Analytics because it can analyse and learn massive amounts of unsupervised data [7]. The core goal of immense statistics analytics is to identify valuable models from large amounts of information which could be utilized for decision-making and prediction [8].

Football matches are one of the most popular types of entertainment, yet they are frequently disrupted by violence between fans or between players. The present difficulty of violence detection in sports data processing is that it is difficult to obtain significant information for classification and notifying security personnel in a short time. Conventional operators viewing surveillance footage respond slowly that results in loss of human life and property. Hence an automated violence detection system which works in a short duration is needed.

This paper proposes a technnique for violation detection system in the football ground that alerts the security persons within a short period of time. Immense statistics analytics and DL will be combined to improve the effectiveness of the violence detection system. Here, the videos from the surveillance cameras are processed using the Flink structure. By using the HOG purpose, the Flink outline divides the edges and retrieves the image frame characteristics. The frames are then labelled on the basis of attributes including Groundside-violence model, Crowdside-violence model, humanoid part model, and non-violence model, are utilized to train the MH-BLSTM for violent scene detection.

Following are the remaining sections of the paper. The literature review is represented in Section II. The proposed technique is represented in Section III, which uses the big data framework namely Apache Flink and Deep learning based Multi head -Bidirectional LSTM. The results and discussion are labelled in Section IV. The conclusion is described in Section V.

2. LITERATURE REVIEW

In 2021, Peixoto et al., [11] presented a method for allowing devices to perceive a top-level notion of violence by dividing into little, more realistic components, including fights, blood, bullets, and explosions, and then combining them afterwards for a clearer idea of the picture. Because of its robust and flexible construction, the detector may be tailored to fit the needs of a wide range of cultures and users. Experiments indicate that the presented technique performs better than existing methodologies.

In 2021, Ullah, et al., [12] provide an operative and resilient method for detecting abnormalities in BVD using AIoT. To assess the effectiveness of their methodology, they run comprehensive experiments on benchmarks constructed on peak of the RWF-2000 and UCF-Crime datasets. In comparison to existing approaches examined across the aforementioned datasets, they claim a 9.88 percent and 4.01 percent improvement in accuracy.

In 2020, Cameron, et al., [13] developed a unique strategy that utilises the object detection algorithm's past detection confidences to build the best independent prioritizing score. A new ensemble technique is also described, which employs a KNN regressor to aggregate the superior of the formerly analyzed measure to build a active prioritizing technique. Using three publicly accessible datasets, this technique is proven to boost the target identification ratio to 60% in comparison to a static sub-sampling baseline.

In 2020, Guedes, and Chávez, [14] provides a technique on the basis of Dynamic Images approach, which employs handmade and CNN features such as the Bag of Visual Words paradigm and an SVM classifier to recognise violent acts including bodily struggle in video streams from literary databases. For the Hockey dataset, the suggested approaches produce an average accuracy of 97.50 percent, 99.80 percent for the Movies dataset, and 93.40 percent for the Crowd dataset. Furthermore, each video's detection of violence was done in hundredths of a second.

In 2021, Wang et al., [15] proposed a brute force detection approach on the basis of integration of convolutional neural networks and trajectory to address the issue of unusual behaviour detection, particularly the poor efficacy and less precision of brute force detection. Using the Hockey and Crow datasets, the study's proposed brute force identification method demonstrated up to 92 percent and 97.6 percent, respectively, accuracy. Experimental data indicates that the violence detection approach proposed in this study improves the video violence detection accuracy.

In 2020, Deepak et al., [16] examined the gradient-lowed attributes' spatiotemporal autocorrelation in order to efficiently identify violent acts in crowded environments. The effectiveness of machine learning algorithms is significantly impacted by the format of the raw data. Then, to detect violent acts in videos, a discriminative classifier is utilised. On comparison with existing methodologies, experimental findings reveal that the suggested methodology outperforms them.

In 2021, Islam et al., [17] propose a 2-stream DL architecture based on Separable Convolutional LSTM (SepConvLSTM) and pre-trained MobileNet, in which 1 stream examines difference of neighbouring frames while the other stream requires in context restrained frames as inputs. On the bigger and more difficult RWF-2000 dataset, their model surpasses the accuracy by more than 2%, while corresponding results of existing method on the lesser datasets. Their results show that the suggested models outperform the competition in terms of computing efficacy and identification accuracy.

In 2018, Mumtaz et al,[18] suggested a deep representation-based model for detection vicious scenarios utilizing the notion of transfer learning to recognize violent human actions. The results indicate that the suggested approach surpasses accuracies of existing methods by learning the most discerning features, which achieve 99.97% and 99.28% accuracies on the Movies and Hockey datasets, appropriately, and by learning the best features for the activity of violent behavior identification in footages.

In 2020, Abdali, and Al-Tuma, [19] suggested a model that comprises of CNN as a spatial feature extractor and LSTM as a temporal relation learning approach with a concentration on the 3-factor model (total prevalence - correctness–quick reaction time). At a frame rate of 131 frames per second, the suggested model achieved an accuracy of 98%. The accuracy and speed of the proposed model were compared to previous research, and it was found that, out of all the approaches already in use for violence detection, the suggested technique had the highest accuracy and the fastest speed.

In 2020, Ehsan, and Mohtavipour, et al., [20] presented a unique Vi-Net architecture on the basis of a deep Convolutional Neural Network (CNN) to identify activities with unusual speed. Optical flow vectors are used to train the Vi-Net network by estimating the movement patterns of objects in the video. They conducted multiple tests on the Hockey, Crowd, and movies datasets, and the results revealed that the suggested architecture outperformed existing approaches relating to accuracy.

3. PORPOSED METHOD

The core factor of the suggested method is to detect the violence in real time for preventing the violence in advance by alerting the security personnel in a short period of time. To succeed in that a violence detection system with better performance have been proposed. Figure 1 represents the work flow of the suggested system.

An input source for the proposed system is a stream of videos from various sources. Video streams are turned into non-overlapping pictures from video streams using the Flink framework. Upon receiving the incoming video streaming block, the Flink framework converts it to frames and passes them to the HOG function in the Flink engine for feature extraction. Flink is faster than other traditional systems in processing streaming blocks in real time. Once the frames have been processed, they are divided into 8 8-pixel cells, with gradient orientations created for every cell. After normalizing the histogram employing surrounding pixels, the features are excerpted from the pictures. According to the activities, the photos are divided into four parts. The "Ground side-violence model" views the fight between the players in the ground. The “crowdside-violence model” contains the physiological harm, maldevelopment or deprivation among the crowd in the audience side. The “Human part model” has representations of autonomous human parts that have been set aside during the violence. The “non-violence model” includes visuals of a background of a specific location where there is no violence.

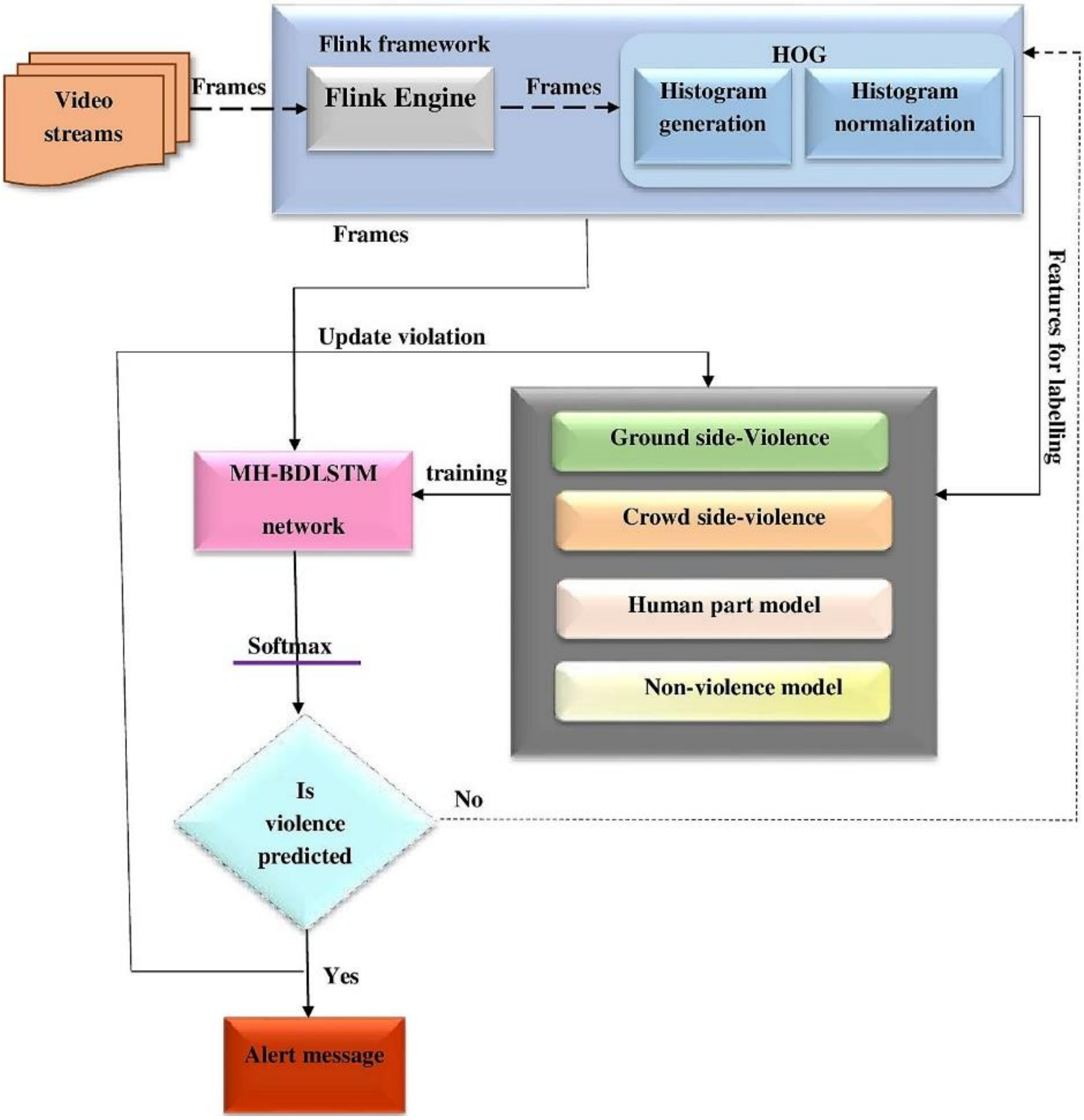


Figure 1. Architecture diagram for the proposed violence detection system

Next, the multihead Bidirectional LSTM (MH-BDLSTM) network is trained using all of the models. Following the training of the network, MH-BDLSTM validates the input frames sequentially and detects violent actions, which can handle both forward and backward dependencies. A multi-head time-dimension approach is used instead of BDLSTM outputs to obtain more valuable space - time information connected to violence detection by reflecting the outputs into other sub-images. A security officer is notified (Groundside Violence, Crowdside Violence) if the input frames match the violent models. Thus, the identified or predicted violent frames in the video are saved in the violence models for later reference. The next subsections will cover the Flink approach, the HOG algorithm, and the MH-BDLSTM neural network in detail.

3.1. Apache Flink

The Flink framework is a distributed processing engine for unbounded and bounded streams of data. Flink is intended to carry out in-memory calculations at any size in all cluster settings. Streams of data are created every time an event occurs. Flink runs any bitstream programme in a data-parallel and pipelined (thus task parallel) fashion. Flink's pipelined runtime technology enables stream processing and bulk/batch programmes to be implemented. Additionally, Flink's runtime natively allows the implementation of repetitive techniques.

Flink data streaming run-time ensures low latency and high throughput with minimum installation requirements. The failure tolerance method in Apache Flink is predicated on Chandy-Lamport distributed snapshots. Because the technique is less weight, it can sustain high throughput rates while also providing excellent consistency guarantees. Flink would'ntsupply its personalinformationrepository system, but it does provide connectors to Amazon Kinesis, HDFS, Apache Cassandra, Apache Kafka, and ElasticSearch.

Flink is the 4G of Big Data. Flink manages memory for clients automatically. Flink works with data at a rapid speed. It can connect to Hadoop and NoSQL databases natively and handle HDFS data. This is the quickest tool for evaluating big data, such as a video stream. The data stream can be processed rapidly and efficiently. Apache Flink, which is based on a pure streaming method, is used to provide good throughput and simplicity of use for sophisticated analysis processing in big data.

3.1.1. Checkpoints, Fault-Tolerance, and Save points

Apache Flink implements a less weight defect tolerance method on the basis of distributed checkpoints. Checkpoints are automatic snapshots of an application's state and a place in a stream. Flink programs with checkpointing configured resume execution from the previous completed checkpoint after a mistake, thereby guaranteeing that application state semantics are preserved. Utilizing hooks exposed by the checkpointing mechanism, external systems can be included in the checkpointing mechanism. Also included in Flink are checkpoints that are physically activated. A user can create a save point, then halt and resume a running Flink programme from the similar application state and location in the stream. Save points allow you to update a Flink programme or a Flink cluster without sacrificing the state of the application. Save points, which were introduced in Flink, now allow users to resume an application with a distinct parallelism, enabling them to adjust to shifting workloads.

Advantages

When compared to other big data technologies, it offers several advantages. The benefits are

- Flink provides APIs that are simpler to develop than MapReduce APIs. It enables for in-memory dispensation, which is significantly quicker. It also adds additional operators to the MapReduce concept, such as join, cross, and union.
- Flink is capable of analysing real-time stream data, graph processing, and machine learning methods. Faster throughput, a shorter latency, and a guarantee that exactly one processing will take place with Flink.
- Flink is also regarded as a viable replacement for Spark and Storm.

3.1.2. API for DataSets

On limited datasets, Flink's DataSet API allows for conversions. There are about 20 distinct types of conversions in the DataSet API. The Application Programming Interface is provided in Scala, Java, and a Python API that is still in development. The DataSet API in Flink is identical to the DataStream API in principle.

3.1.3. SQL and the Table API

The Table API in Flink's Scala and Java DataStream and DataSet APIs provides a SQL-like expression language for batch processing and relational stream. A relational Table abstraction is used by the Table API and SQL interface. External data sources, as well as existing DataStreams and DataSets, can be used to generate tables. On Tables, relational operations include selection, joins, and aggregation are supported via the Table API.

Regular SQL may also be used to query tables. Table API and SQL are functionally comparable and may be used together in the same software. The logical plan, that was specified by SQL queries and relational operators, is optimised employing Apache Calcite and changed into a DataSet or DataStream programme once a Table is transferred back into a DataStream or DataSet.

3.1.4. Flink streaming

Apache Flink uses the Kappa construction. The core impression overdue the Kappa design is to use a single stream processing engine to manage together group and real-time statistics. The Lambda architecture, which includes distinct processors for batch and streaming data, is used by the majority of big data frameworks. Architecture of framework shown in Figure 2.

Batch dispensation refers to the processing of large amounts of data in a single batch over a particular timeframe. Continuous streams of data can be instantly processed with stream processing. Data streams can be handled by Flink

Streaming in real-time, utilizing the pipelined Flink engine and include customizable windows. The feature extraction can be carried out employing HOG function.

3.1.5. Features of Flink

- Using the DataStream API, Flink programmes enable users to execute business logic requiring a contextual state while processing data streams at any scale, resulting in stateful streaming at any scale.
- Flink provides a defect-tolerance method on the basis of asynchronous checkpointing and periodic
- Exactly-Once Consistency: In the event of a failure, the Flink core guarantees that every event in the stream is supplied and worked precisely once.
- Scalability: programmes are parallelized so that the number of processing jobs may be increased or decreased.
- Flink apps use local, typically in-memory, state to complete all calculations, resulting in very less processing delays.
- Flink connects to a wide range of data sources, such as Elasticsearch, Apache Cassandra, Apache Kafka, Kinesis, and many others.
- Deployment options: Flink is compatible with a variety of cluster setups, including YARN, Apache Mesos, and Kubernetes.
- A library for detecting patterns in data streams using Complex Event Processing (CEP).
- Java and Scala have fluid APIs.
- Flink is a genuine streaming engine, as opposed to Spark Streaming's micro-batch processing paradigm.

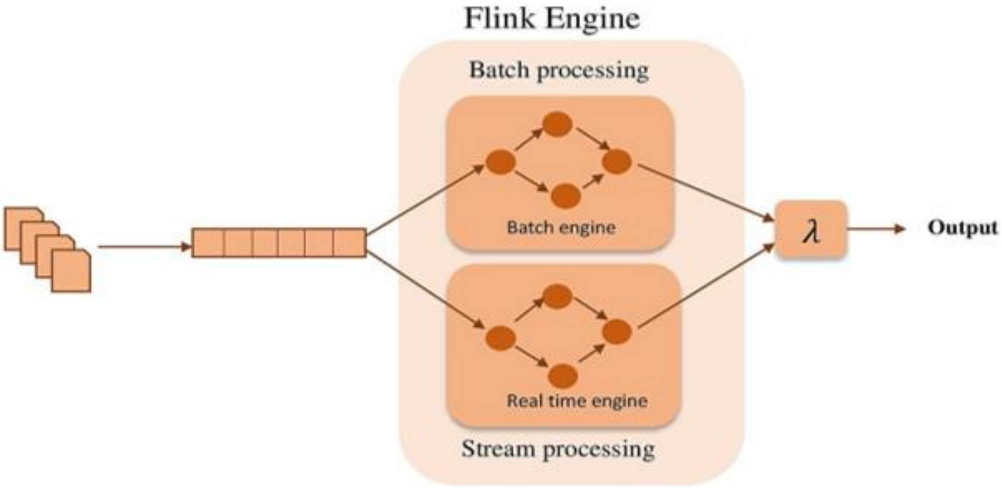


Figure 2. Architecture of Flink framework

3.2. Histogram of Oriented Gradients (HOG) function

The histogram of directed gradients is used in computer vision and image processing to identify objects. Gradient orientations are counted based on the number of times they appear in a particular section of a photograph. This approach uses overlapping local contrast normalisation and is based on a dense grid of evenly spaced cells, which sets it apart from edge orientation histograms, shape, and contextual scale-invariant feature transform descriptors.

This histogram describes the aspect and form of local objects inside an image by measuring the delivery of gradients or edge directions. After the frame is splitted into slight corresponding pieces known as cells, a histogram of gradient oeientations for each pixel is constructed. These histograms make up the descriptor. The function of HOG will be carried out in 4 stages. They are described as follows:

3.2.1. Gradient Calculation

The calculation of the gradient values is the initial stage in the process. The gradient is calculated by summing the angle and magnitude of the picture. First, the Gaand Gb values are found for each pixel in a 3x3 pixel block. Ga and Gb are initially calculated for each pixel value using the following formulas.

$$G_a(R, C) = I(R, C+1) - I(R, C-1) \tag{1}$$

$$G_b = I(R-1, C) - I(R+1, C) \tag{2}$$

where R and C stand for rows and columns, respectively. Once Ga and Gb have been established, the magnitude and angle of each pixel are computed using the following formulas.

$$Mag(\varphi) = \sqrt{G_a^2 + G_b^2} \tag{3}$$

$$Angle(\theta) = |\tan^{-1}(G_b/G_a)| \tag{4}$$

3.2.2. Orientation binning

The second step in the procedure is to create cell histograms. Each pixel in the cell represents a weighted vote for an orientation-based histogram channel, depending on the principles that were discovered during the gradient computation. The cells of a histogram can be rectangular or radial, and the channels are evenly spaced over a range of 0 to 360 degrees or 0 to 180 degrees, depending on whether the gradient is signed or unsigned. For every block, an n-point histogram is produced. Using an n-point histogram, n bins with θ-degree angle ranges are produced in the histogram.

$$Number\ of\ bins = n(ranging\ from\ 0^o\ to\ 180^o) \quad (5)$$

$$step\ size(\Delta\theta) = 180^o/Number\ of\ bins \quad (6)$$

3.2.3. Descriptor blocks

The gradient capacities must be locally adjusted to accommodate for differences in light and contrast, that requires combining the cells collectively into bigger, geographically linked blocks. Circular C-HOG blocks and Rectangular R-HOG blocks are the two most common block geometries.

3.2.4. Block normalization

A normalisation process reduces the negative impact of differences in contrast between images of the same thing. In general, the gradient is affected by lighting. When we divide or multiply pixel terms by a value to attain lighter or darker, the gradient magnitude and histogram values alter. It is important that histogram values remain unchanged regardless of lighting conditions. Within a block, the histogram vector v is normalized. One of the following standards could be suitable:

- L1 norm
- L2 norm
- L2-Hys (Lowe-style clipped L2 norm)

Let fyj be a non-normalized vector that contains all the histograms in a particular block. The L2 norm is used to standardize the fy values for each block:

L2 norm:

$$f_{yj} \leftarrow \frac{f_{yj}}{\sqrt{\|f_{yj}\|^2 + s}} \quad (7)$$

Where s is the minimum worth, additional to the square of fyj in order to evade zero separation error.

$$t = \sqrt{y_1^2 + y_2^2 + \cdots y_n^2} \quad (8)$$

$$f_{yj} = \left[\left(\frac{y_1}{t}\right), \left(\frac{y_2}{t}\right), \dots \left(\frac{y_n}{t}\right)\right] \quad (9)$$

A video stream patch is divided into eight nonoverlapping pixels, or cells, to form this study. The gradients for each pixel are determined in these cells. The surrounding cells can be used to standardize this histogram. This will increase the robustness of the texture and lighting variations. The edges are labeled in agreement with the Groundside-Violence Model, Crowdside Violence Model, Human Part Model, and Non-violence Model after the features have been extracted by employing the HOG function. The MH-BDLSTM network is then used to train the model.

3.3. Multi Head Bidirectional LSTM

For the violence detection model, a hybrid modelon the basis of bidirectional LSTM and multi-head attention mechanism was developed. The framework is depicted in the image below (Figure 3). The suggested model structure is categorized into four sections.

- Extract the text's word embedding
- Upload the word vector to the BiLSTM framework.
- Implement a Multi-head Attention method to acquire important data from several subspaces and emphasise the significance of various aspects;
- For detection of violence, use the softmax layer.

3.3.1. Word vector representation

Words are at the bottom of the structure, with a subtext X composed of n number of words being represented as (y1, y2, y3..., yn-2, yn-1, yn), and the input layer as:

$$I = [wi_1, wi_2, wi_3, \dots, wi_{m-2}, wi_{m-1}, wi_m] \in R^{m\times d} \quad (10)$$

3.3.2. BidirectionalLSTM

The LSTM is a sort of RNN which could learn and remember long-term dependencies without experiencing vanishing gradient descent or explosion difficulties. The sequence's future data is recorded in the backward layer, and its past data is kept in the forward layer. Both layers share the same output layer. The multi-head technique is used in this network, which leverages bidirectional LSTM. Multi-head attention permits the forms to concurrently assist to intake from various representation subspaces at several places.

3.3.3. Layer of word encoder

When the words are expressed using word embedding W, each word in the comment C is independent of the other words. A new representation for each word is created in this layer by combining contextual data from both directions in a comment. A bidirectional LSTM is made up of a forward LSTM H which discusses feedback from c_1 to c_n.as well as a backward LSTM $\overleftarrow{H}$that reads the comment from $c_n$toc_1.

$$\vec{H}_M = \overrightarrow{LSTM}(W_m, \overrightarrow{H_{m-1}}) \quad (11)$$

$$\overleftarrow{H_M} = \overleftarrow{LSTM}(W_m, \overleftarrow{H_{m+1}}) \quad (12)$$

To derive hidden state representation H_m for each word W_m, simply concatenate forward hidden state $\vec{H}$ and backward hidden state $\overleftarrow{H}$, i.e. H_m = [$\overrightarrow{H_m}, \overleftarrow{H_m}$]. This method aids in the capture of information from the entire sentence around each word W_m. $H \in R^{N\times 2p}$ denotes all the hidden states of the words W_m, with $\vec{H}$and $\overleftarrow{H}$ being s in size.

$$H = (H_1, H_2, \dots H_n) \quad (13)$$

3.3.4. Multihead attention

MHAT is an improved version of the classic attention mechanism that also outperforms it. Several factors can influence how a frame is paid attention, requiring the use of multiple heads of attention, in which each frame is assigned appropriate weight based on multiple aspects to convey the overall semantics of the statement. One head is calculated at a time in this method. It's important to remember that there are h times to accomplish this, that is known as multi-head, but the parameters W for every linear relationship of Q, K, and V are distinct. In the following step, all m times scaled dot-product attention outcomes are combined, and the linearly transformed value is utilised as the MHAT result.

$$H_1 = attention(QW_j^Q, KW_j^K, VW_j^V) \quad (14)$$

$$M(Q, K, V) = Concatenate(H_1, \ldots . H_n)W^P \quad (15)$$

This method explores the inner connections of sentences using self-attention, here K = V = Q. A weight matrix M and a feature representation frep are generated by this MHAT process.

$$y_i = \tan h(W_r H_i + a_r) \quad (16)$$

$$\mathrm{f_{rep}} = \mathrm{Multihead(Y, Y, Y)} \quad (17)$$

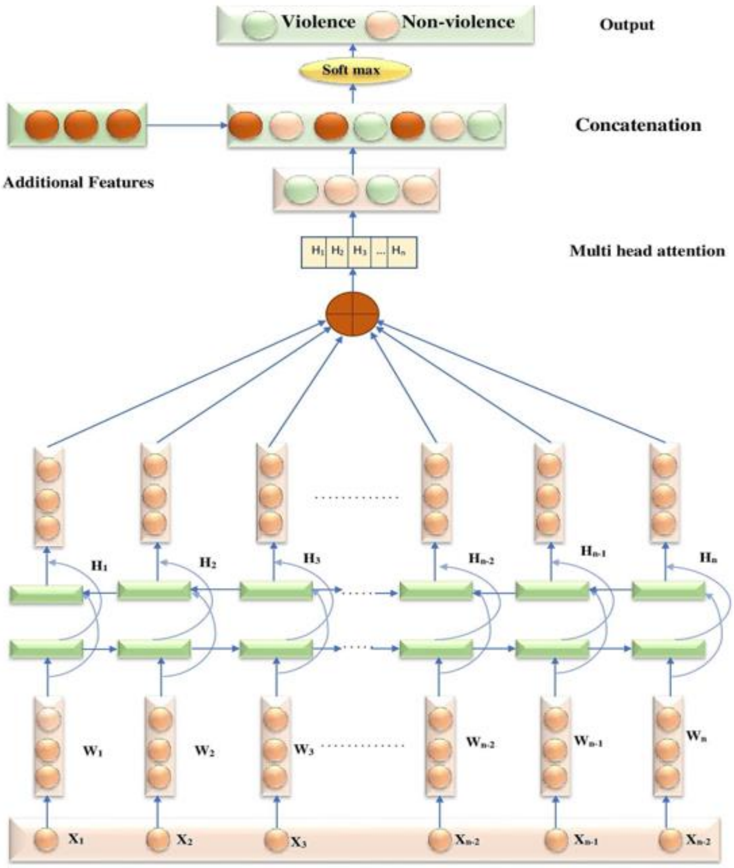


Figure 3. Architecture of Flink framework

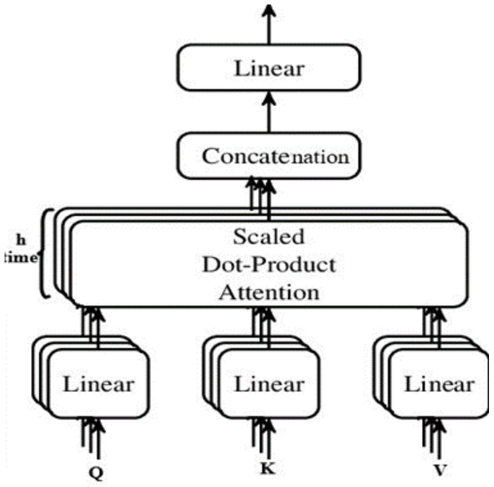


Figure 4. Structure of multihead attention

For violence detection, the resultant vector is routed to the softmax layer. The following is the outcome of the detection:

$$\widehat{z} = softmax(Wz_{gap} + a) \quad (18)$$

The goal of introducing cross-entropy is to assess the types, that shows the difference between the detected violence types y and the detected non-violence types. Structure of multihead attention is shown in Figure 4.

$$l = -\sum_j Z_j \log \widehat{z_j} \quad (19)$$

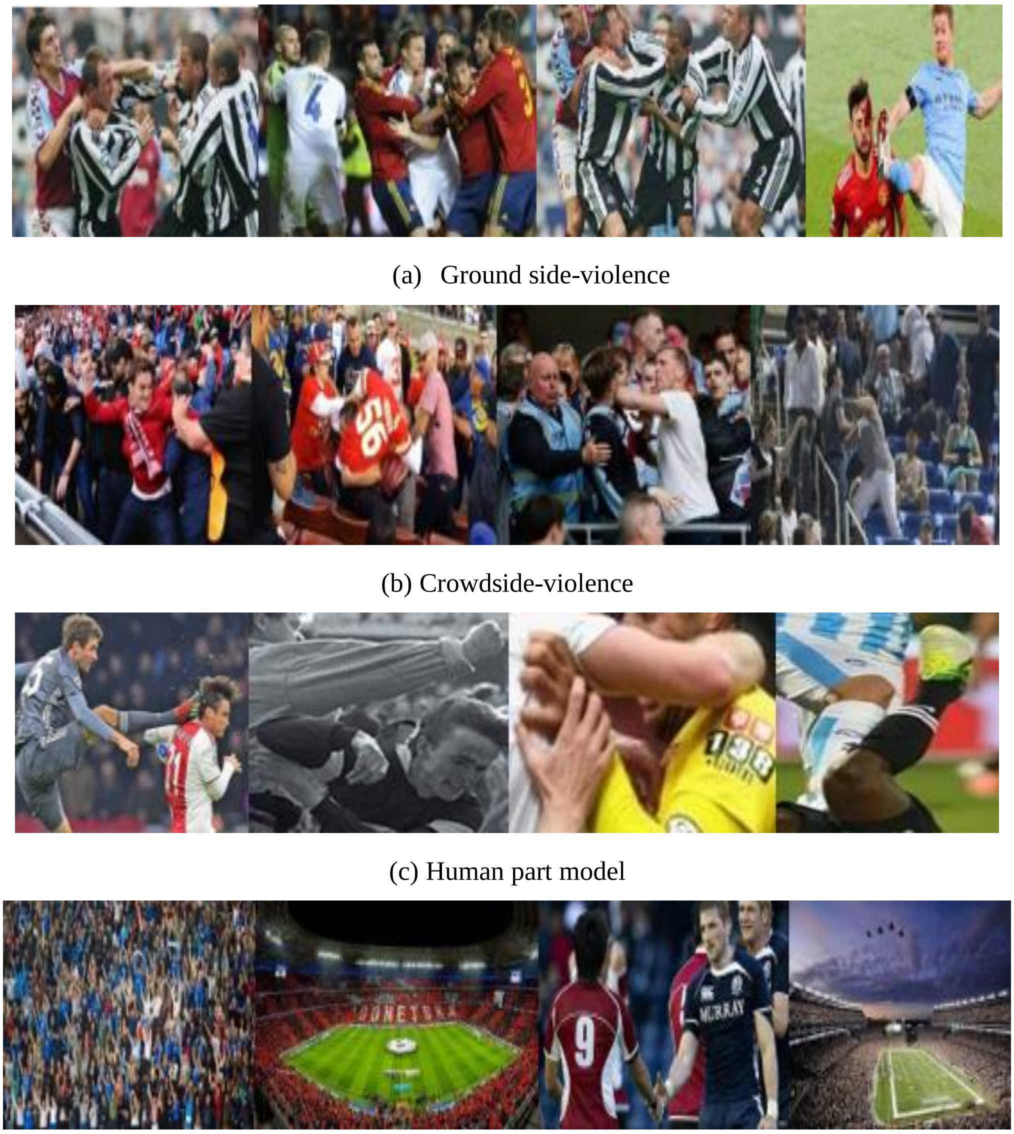
where j is the sentence's index number.

4. RESULTS AND DISCUSSIONS

This part discusses how effective the suggested violence detection system is in identifying violent views in real-time footage. It was designed to detect violence in football stadiums where there is a possibility of violence between players or audience.

Dataset

The dataset is based on the new RWF2000 (Real-World Fighting) dataset from the YouTube website, which contains 2,000 clipped video clips acquired by surveillance cameras from real-world settings. There are 2,000 video clips in the dataset, which are divided into two parts: Half of the videos depict violent acts, while the other half depict non-violent activities. Representation of 4 types of models is shown in Figure 5.



(a) Ground side-violence

(b) Crowdside-violence

(c) Human part model

(d) Non-violence model

Figure 5. representation of 4 types of models

4.1. Performance Evaluation

A comparison is made between the suggested approach and four other methods: Convolutional Neural network Bidirectional LSTM (CNN-BDLSTM), Seperable convolutional LSTM and pre trained mobilenet (SepConvLSTM-M), 3D Convolutional Neural Network with a Support Vector Machines (SVM) classifier(C3D), ResNet50CNN

The network is trained using the retrieved features after extracting them with the HOG function and establishing the MH-BDLSTM parameters. Validation of the violence detection system is performed by calculating precision and recall values. Precision refers to the similarity of two or more measurements to one another. Precision is the word given to a positive predictive value. It's the percentage of retrieved occurrences that are tightly related.

$$Precision(P) = \frac{True\ positive\ rate}{True\ positive\ rate + False\ positive\ rate} \tag{18}$$

Table 1. Comparison of precision for violence detection

Models	Precision			
	videos from dataset		**Real-Time**	
	Violence	Non-violence	Violene	Non-violene
CNN-BDLSTM	97.5	96.5	89.2	87.57
SepConvLSTM	97.65	97	89.5	88.1
C3D	97.71	97.54	89.65	88.96
ResNet50 CNN	97.95	97.87	89.72	89.02
Proposed method	98.6	97.96	90.1	90

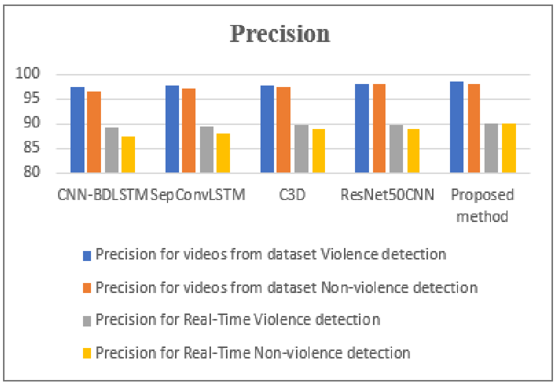


Figure 6. Graphical representation of comparison of precision

Figure 6 shows the precison findings of five different approaches. The proposed violence detection method acquires higher precision results of 90.1% for Real time, whereas other methods such as CNN-BDLSTM, SepConvLSTM-M, C3D, and ResNet50CNNprovide precision results of 89.2 percent, 89.5 percent, 89.65 percent, and 89.72 percent, respectively, as shown in Table 1.

Table 2. Comparison of accuracy for violence detection

Models	Accuracy			
	videos from dataset		Real-Time	
	Violence	Non-violence	Violence	Non-violene
CNN-BDLSTM	95.5	93.5	87.2	86.57
SepConvLSTM	95.1	94	88.5	88.1
C3D	95.51	95.4	89.25	88.96
ResNet50CNN	96.95	96.87	89.52	89.02
Proposed method	98.6	97.96	90.1	90

Figure 7 shows the accuracy findings of five different approaches. The proposed violence detection method acquires higher accuracy results of 90.1% for Real time, whereas other methods such as CNN-BDLSTM, SepConvLSTM-M, C3D, and ResNet50CNNprovide precision results of 87.2 percent, 88.5 percent, 89.25 percent, and 89.52 percent, respectively, as shown in Table 2.

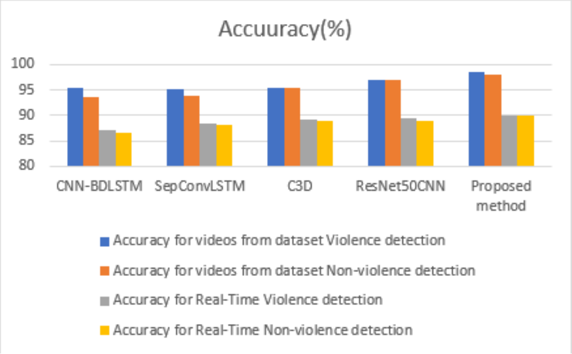


Figure 7. Graphical representation of comparison of Accuracy

Table 3. Comparison of Recall for violence detection

Models	Recall			
	videos from dataset		Real-Time	
	Violence	Non-violence	Violene	Non-violence
CNN-BDLSTM	92.5	91.2	85.2	84.57
SepConvLSTM	92.45	92.4	85.76	85.68
C3D	93.51	93.4	86.25	85.96
ResNet50CNN	94.65	94.17	86.52	86.02
Proposed method	95.42	95.26	87.65	86.90

Recall is the term for sensitivity. The recall is the percentage of relevant examples that might be retrieved that is greater than the entire number of relevant instances. Each accuracy and recall are discussed below based on an understanding and measurement of significance.

$$Recall(R) = \frac{True\ positive\ rate}{True\ positive\ rate+False\ negative\ rate} \tag{19}$$

Figure 8 shows the recall findings of five different approaches. The proposed violence detection method acquires higher recall results of 87.65% for Real time, whereas other methods such as CNN-BDLSTM, SepConvLSTM-M, C3D, and ResNet50CNNprovide precision results of 85.2 percent, 85.76 percent, 86.25 percent, and 86.52 percent, respectively, as shown in Table 3.

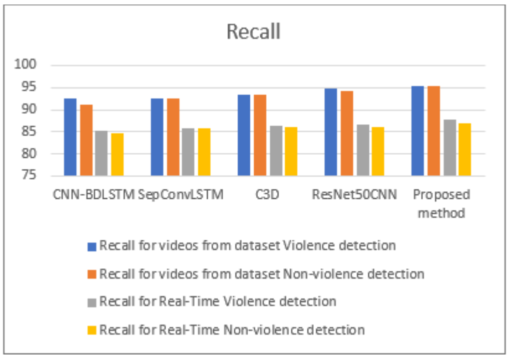


Figure 8. Graphical representation of comparison of Recall

Table 4. Comparison of accuracy for violence detection

Models	Time for execution			
	videos from dataset		Real-Time	
	Violence	Non-violence	Violence	Non-violence
CNN-BDLSTM	25	27	29	30
SepConvLSTM	23	24	26	27
C3D	20	21	23	25
ResNet50CNN	16	18	20	21
Proposed method	11	12	14	15

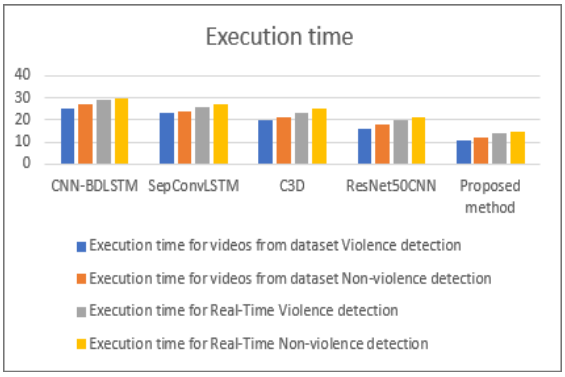


Figure 9. Graphical representation of comparison of execution time

Figure 9 shows the execution time findings of five different approaches. The proposed violence detection method acquires less execution time of 15% for Real time, whereas other methods such as CNN-BDLSTM, SepConvLSTM-M, C3D, and ResNet50CNNprovide precision results of 29 percent, 26 percent, 23 percent, and 20 percent, respectively, as shown in Table 4.

5. CONCLUSION

The rate of violence in football matches has risen dramatically in recent years, whether among players or fans. Security personnel have to be notified in real-time to prevent violence from occurring. Using Flink, the HOG function helps to identify violent behavior by extracting information from video frames. Based on the retrieved features, we can classify the frames into four categories: groundside-violence, crowdside-violence, human part model, and non-violence model. Multihead bidirectional LSTM networks are then trained using the four models to identify violent frames in videos. Compared to a single LSTM, this network arrangement converges faster. When compared with existing systems, the proposed violence detection methodology significantly increases accuracy upto 1.6453%, precision upto 0.646%, recall upto1.959%, and reduces execution time upto 60% than other existing methods. In the future, further research will be conducted to find out if the proposed methodologies can be used to address concerns regarding violence detection.

CONFLICTS OF INTEREST

The author proclaims that there are no conflicting interests to disclose in this study.

FUNDING STATEMENT

Not applicable.

ACKNOWLEDGEMENTS

The author would like to express his heartfelt gratitude to the supervisor for his guidance and unwavering support during this research for his guidance and support.

REFERENCES

[1] F. U. M. Ullah, A. Ullah, K. Muhammad, I. U. Haq, and S. W. Baik, "Violence detection using spatiotemporal features with 3D convolutional neural network", *Plastics*, vol. 19, no. 11, pp. 2472, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[2] E. Fenil, G. Manogaran, G. N. Vivekananda. Thanjaivadivel, S. Jeeva, and A. Ahilan, "Real time violence detection framework for football stadium comprising of big data analysis and deep learning through bidirectional LSTM", *Computer Networks*, vol. 151, pp. 191-200, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[3] A. Mumtaz, A.B. Sargano, and Z. Habib, "Violence detection in surveillance videos with deep network using transfer learning", *2nd European Conference on Electrical Engineering and Computer Science (EECS)*, pp. 558-563, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[4] M. Rai, A. A. Husain, T. Maity, and R. K. Yadav, "Advance intelligent video surveillance system, (AIVSS): a future aspect", In Intelligent Video Surveillance. Intech Open, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[5] B. N. Subudhi, D. K. Rout, and A. Ghosh, "Big data analytics for video surveillance", *Multimedia Tools and Applications*, vol. 78, no. 18, pp. 26129-26162, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[6] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "A survey on deep learning for big data", *Information Fusion*, vol. 42, pp. 146-157, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[7] M. M. Najaf Abadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald and E. Muharemagic, "Deep learning applications and challenges in big data analytics", *Journal of big data*, vol. 2, no.1, pp. 1-21, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[8] B. Jan, H. Farman, M. Khan, M. Imran, I. U. Islam, A. Ahmad, S. Ali and G. Jeon, "Deep learning in big data analytics: a comparative study", *Computers & Electrical Engineering*, vol.75, pp. 275-287, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[9] W. Ullah, A. Ullah, I. U. Haq, K. Muhammad, M. Sajjad and S.W. Baik, "CNN features with bi-directional LSTM for real-time anomaly detection in surveillance networks", *Multimedia Tools and Applications*, vol. 80, no. 11, pp. 16979-16995, 2021.[CrossRef] [Google Scholar] [Publisher Link]

[10] J. Li, X. Jiang, T. Sun, and K. Xu, "Efficient violence detection using 3d convolutional neural networks", *16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1-8, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[11] B. M. Peixoto, B. Lavi, Z. Dias, and A. Rocha, "Harnessing high-level concepts, visual, and auditory features for violence detection in videos", *Journal of Visual Communication and Image Representation*, pp. 103174, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[12] W. Ullah, A. Ullah, T. Hussain, K. Muhammad, A. A. Heidari, J. Del Ser, S. W. Baik, and V. H. C. De Albuquerque, "Artificial Intelligence of Things-assisted two-stream neural network for anomaly detection in surveillance Big Video Data", *Future Generation Computer Systems*, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[13] J. Cameron, M. E. Kaye, and E. Scheme, "Dynamic prioritization of surveillance video data in real-time automated detection systems", *Expert Systems with Applications*, vol. 161, pp.113672, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[14] A. R. M. Guedes and G. C. Chávez, "Real-Time Violence Detection in Videos Using Dynamic Images", *In XLVI Latin American Computing Conference (CLEI)*, pp. 503-511, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[15] P. Wang, P. Wang, and E. Fan, "Violence detection and face recognition based on deep learning", *Pattern Recognition Letters*, vol. 142, pp. 20-24, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[16] K. Deepak, L. K. P. Vignesh and S. Chandrakala, “Autocorrelation of gradients-based violence detection in surveillance videos”, ICT *Express*, vol. 6, no. 3, pp. 155-159, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[17] Z. Islam, M. Rukonuzzaman, R. Ahmed, M. Kabir and, M. Farazi, “Efficient Two-Stream Network for Violence Detection Using Separable Convolutional LSTM”. arXiv preprint arXiv:2102.10590, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[18] A. Mumtaz, A. B. Sargano, and Z. Habib, “Violence detection in surveillance videos with deep network using transfer learning”, *2nd European Conference on Electrical Engineering and Computer Science (EECS)*, 2018, pp. 558-563, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[19] A. M. R. Abdali, and R. F. Al-Tuma, “Robust real-time violence detection in video using cnn and lstm”, 2nd Scientific Conference of Computer Sciences (SCCS) 2019, pp. 104-108, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[20] T. Z. Ehsan, and S. M. Mohtavipour, “Vi-Net: A Deep Violent Flow Network for Violence Detection in Video Sequences”, *11th International Conference on Information and Knowledge Technology (IKT*), 2020, pp. 88-92, 2020. [CrossRef] [Google Scholar] [Publisher Link]

AUTHORS

M. Dhipa working as Associate Professor in the Department of Biomedical Engineering at Nandha Engineering College, Erode, Tamil Nadu, India. She completed B.E (EIE) in Easwari Engineering College, Madras University, Chennai in 2004 and M.E (Applied Electronics) in K. S. R. College of Technology, Anna University, Chennai in 2006. She pursued Ph.D. under Anna University, Chennai, Tamil Nadu. She is having 16 years of teaching experience in various institutions. She has published about 12 papers in various International Journals. Her area of interest includes wireless networks.

D. Anitha working as Associate Professor in the Department of Information Technology at Muthayammal Engineering College, Namakkal, Tamil Nadu, India.

Arrived: 18.11.2023

Accepted: 20.12.2023

International Journal of Data Science and Artificial Intelligence (IJDSAI)
Volume 01, Issue 02, November – December (2023)

RESEARCH ARTICLE

DINGO OPTIMIZED FUZZY CNN TECHNIQUE FOR EFFICIENT PROTEIN STRUCTURE PREDICTION

P. G. Sreelekshmi [1], * and S. C. Ramesh [2]

[1] Lecturer in Computer science, Department of BCA, University Institute of Technology, Malayinkeezhu Under University of Kerala, India

[2] Associate Professor, Department of Electronics and Communication Engineering, PSN College of Engineering and Technology, Tirunelveli, India.

Corresponding e-mail: Sreelekshmi.prf12@outlook.com

Abstract – Protein is made up of a variety of molecules that are required by living organisms, such as enzymes, hormones, and antibodies. In step 2, the max-pooling layer and the convolutional layer evaluate the input data to create the finest feature map F1, which is half the image size in both horizontal and vertical directions. The full feature is then retrieved in step 2 using the max pooling layer and the residual block at the proper resolution. In this paper, we introduce Di-Fuzzy CNN (Fuzzy Convolutional Neural Network with Dingo optimizer), a novel technique for predicting protein activities that incorporates two types of information they are protein sequence and protein structure. We extract diverse features at different scales utilizing convolutional neural networks to provide comprehensive information for feature segmentation. To handle a variety of uncertainties in feature selection and produce segmentation results that are more dependable, fuzzy logic modules are employed. Finally, we employ Dingo optimization to boost the suggested method's effectiveness and speed in order to produce the best outcomes. Using a variety of datasets, the suggested model has been tested (HSSP, PDB, UGR14b, DSSP). Tests demonstrate that our approach can decrease FPR, increase protein structure accuracy, decrease prediction time, and increase TPR for feature selection. Our predictive model performs better than most state-of-the-art techniques.

Keywords *–Amino Acid Features, Protein Structure, Convolutional Neural Network (CNN), Fuzzy logic, Dingo Optimization algorithm.*

1. INTRODUCTION

Polymeric macromolecules known as proteins are made up of linear chains of amino acid building blocks connected by peptide bonds. Different biological mechanisms in living things produce proteins. [1] The structure of a protein and the chemical characteristics of its amino acids determine its activity. The genetic sequence of a protein can be used to determine its structure. Additional information about protein structure can be obtained by predicting the primary, quaternary, secondary, and tertiary structures [2]. Put differently, protein structure prediction refers to the process of estimating a protein's three-dimensional structure from its fundamental structure [3].

Protein structure prediction is the process of forecasting a protein's different amino acid sequences from its three-dimensional structure [4]. Its folding, secondary, tertiary, and quaternary structures may all be predicted from its fundamental structure. The issues of protein design and protein structure prediction are essentially different. One of the main objectives of theory and bioinformatics in medicine (e.g., drug design) is the prediction of protein structure.

The basic structure of the amino acid series is depicted in Figure 1. A protein's matching gene determines its main structure. α-helix could be the typical secondary structural state. Since hydrogen bonds develop within the chain, they are entropically benifical than beta sheets, despite the fact that their potential energy is not as low as that of beta sheets. The three-dimensional structure of a protein is mostly determined by the interactions between the R groups of the amino acids that make up the protein. Proteins must be sampled in various experimental conditions in order to identify the quaternary structure of proteins, which can be done using a range of experimental techniques [5], [18].

For the purpose of estimating the three-dimensional structure, protein structure prediction is crucial. There is a widespread misperception that it is hard to infer a protein's structure from its amino acid sequence since the amino acid sequence contains sufficient information to reveal a protein's three-dimensional structure. Extrapolating characteristics from amino acid sequences is a crucial step in increasing the precision of protein structure prediction [6, 7]. Utilizing sizable protein databases, protein structure prediction ascertains if a query sequence, in whole or in part, resembles a known structure [8], [19].

Finding a protein's structure from a collection of amino acids is a difficult task in molecular biology and bioinformatics. Many studies use different data mining techniques to predict the structure of proteins. Nevertheless, the computing time and forecast accuracy of earlier methods were inadequate [20]. For over ten years, protein structures have been predicted through the application of neural networks. Inspired by neural networks' recent success, DL networks have been utilized in several articles to predict protein shapes. We suggested a Di-Fuzzy CNN to improve the accuracy and speed of protein structure prediction in order to solve the current problems [21-24].

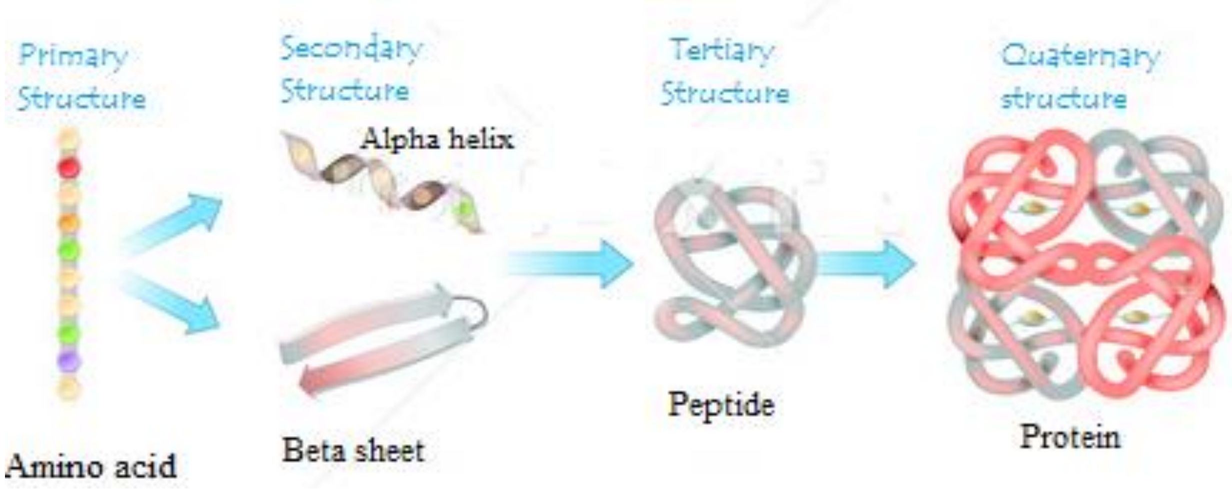


Figure 1. Resultant Graph of the Proposed System

The creation and application of protein structure predictions is explained in this article. Section II discusses related studies on protein structures that make use of different deep learning approaches. Section III presents the proposed Di-Fuzzy CNN (Dingo Fuzzy Convolutional Neural Network), along with a description and associated algorithms. Section IV contains performance results and associated analysis. Section V concludes with additional work and conclusions.

2. LITERATURE SURVEY

In 2020 Giri, et al [9] proposed, a MultiPredGO a novel multimodal technique for predicting protein functions using two separate types of information like protein secondary structure and protein sequence. We evaluated our findings to a variety of unimodal approaches in addition to two multi-modal protein function prediction methods like DeepGO and INGA. Our proposed method achieves better score. For cellular component and molecular comparisons, there were 13.05 percent and 30.87 percent enhancements over the best available comparing method.

In 2020 Zhou et al., [10] proposed, a monitored learning technique known as combining deep neural networks (CDNN) for protein structure prediction. The RMSProp optimizer uses the crossentropy error to educate the CDNN architecture. With multiple CNNs, the suggested design may remove amino acid data and mix them with raw features to train massive LSTM networks. In comparison to other approaches, the output suggest that the suggested techniques can achieve reasonable enactment with the suitable parameters.

In 2020 Bingzhen, et al., [11] suggested utilizing a confusion matrix to choose a random forest classification model. Using the "remove poor models" strategy, the forest models are selected at random. In three various data sets, the new outcomes reveal that the new technique has greater average classification accuracy and stability than the prior iteration. As a result, the confusion matrix-based random forest image classification model can increase random forest classification ability.

In 2019 Akter, and Holder, [12] proposed, a graphical feature-based framework that derives graphical features from sensor network data and employs feature selection approaches to choose more valuable features for such a classifier to have in prediction problems. Using movement data from smart home motion sensors and mobile phone GPS sensor information, as well as demographic data from smartphone GPS sensor data, the researchers used the suggested approach to forecast activity. additionally forecast. We discovered that when non-graph-based features are added to graphical feature-based frameworks, the outcomes improve.

In 2019 Gao et al., [13] proposed, A novel method for predicting equilibrium contacts in proteins is called Deep Structural Inference for Proteins (DESTINI), which blends template-based structural models with DL algorithms. DESTINI accurately predicts the tertiary structure 4 times for "hard" targets while simultaneously improving model quality for "easy" targets. DESTINI's much improved performance is partly due to the introduction of improved contact prediction template model. This paper outlines a viable technique for resolving the prediction problem in protein structure.

In 2018 Zamil, and Rahman, [14] proposed a multiscale local descriptor (MLD) to retrieve multiscale local information by feature extraction from a set of proteins. Decision trees, random forests, and bootstrap aggregation are used in classification approaches. Several algorithms produce diverse outcomes, and ensemble classification performs more accurately than current techniques. It is discovered that random forests and bootstrap aggregation are at least 10% more accurate than decision tree methods.

In 2018 Yavuz, et al., [15] proposed, An MLP methodology for predicting the secondary structure of proteins. The amino acid sequence was used to estimate protein secondary structure. There are two steps to the classification: MLP and direct MLP with CSA enhancement. The success rate for direct MLP categorization is 84.01 percent. For various numbers of rounds and hidden layers, the MLP with CSA arrangement achievement is investigated. To summarise, using CSA prior to categorization is advised for better prediction accuracy.

In 2018 Xie, et al., [16] proposed, A fuzzy support vector machine for secondary structural identification for predicting the amino acid features. Agreeing to the K-nearest neighbour algorithm, hyperplanes are assigned big membership values, whereas outliers are assigned small membership values. To test this strategy, we employed 3 databanks (e.g., CB513, data11996and RS12). Overall, our results for secondary structure prediction are better than regularly used approaches.

In 2017 Wang et al., [17] proposed, A deep recurring encoding-decoder network, secondary structure recurrent encoder. suggested using a decoder network. The CB513 and CullPDB public datasets are used to test the suggested model. Especially well-suited to modeling sequence and structural links between input protein attributes and secondary structure are encoder/decoder designs used in conjunction with GRUs. It also performs better than the opposition in terms of Q8 and Q3 accuracy. We outperformed earlier methods in predicting Q3 and Q8 with 68.20 percent and 73.1 percent accuracy in fewer epochs on the CB513 and CullPDB datasets.

In 2017 Liu, et al., [18] proposed, two-dimensional deep convolutional neural network, the protein's secondary structure was predicted. Based on a two-dimensional input matrix, two-dimensional CNNs are better at extracting sequence interaction features and storing unique amino acid position data. Our predictive model performs better than most state-of-the-art techniques.

3. PROPOSED METHOD

A crucial step in theoretical chemistry and bioinformatics is the prediction of protein structures. Because predicting protein structure is a critical procedure in medicine (for example, medication design) and biotechnology. A large number of research projects are currently underway with the goal of determining the protein structure using various classification techniques. However, existing categorization techniques performed poorly. The Di-Fuzzy CNN Technique was created to address these restrictions.

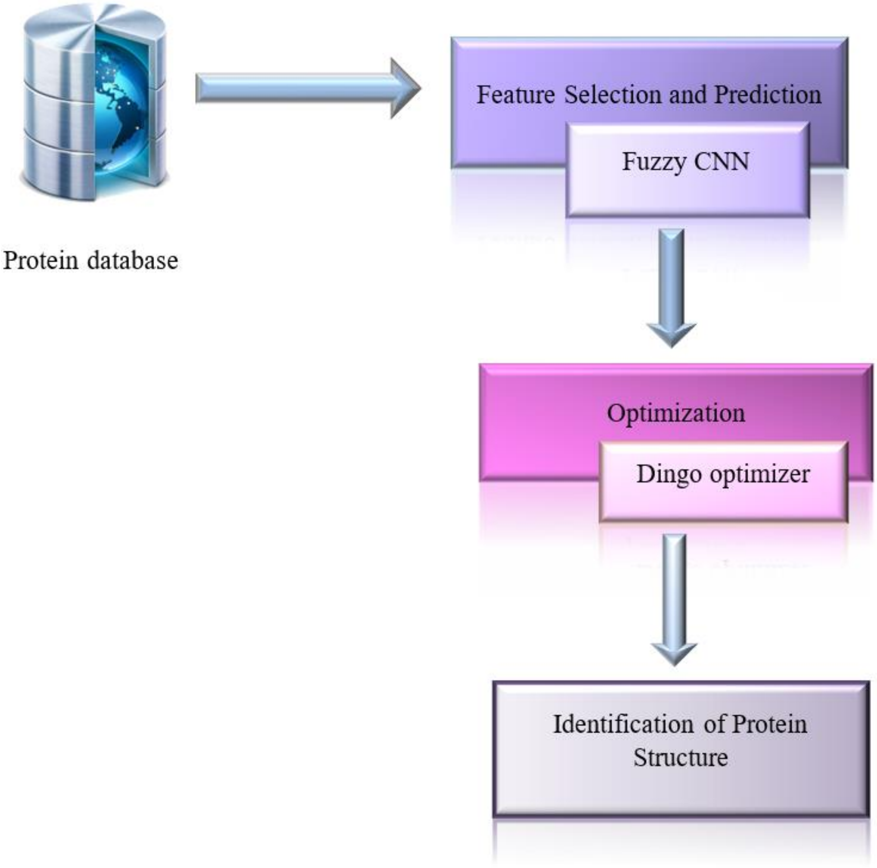

Protein database
Feature Selection and Prediction
Fuzzy CNN
Optimization
Dingo optimizer
Identification of Protein Structure

Figure 2. Proposed Methodology

Figure 2 depicts the overall process of using the Di-Fuzzy CNN technique to predict protein structure more efficiently and in less time. In the above figure, Fuzzy CNN technique is used to identify relevant amino acid characteristics from a large protein dataset and predict the protein structure. The dingo optimizer algorithm improves the accuracy and reduces the computational time as a result.

3.1. Feature Selection and Prediction

Convolutional neural networks and fuzzy learning are combined to form a new neural network structure for selection and prediction called as Fuzzy Convolutional Neural Network (Fuzzy CNN). The proposed technique uses fuzzy learning to deal with protein structure uncertainty, and the fuzzy logical units are smoothly combined into the Neural Network. Each fuzzy logical units are guided by a feature map at a specified balance, with the goal of establishing a link between the segmentation and the features outcome. A robust and accurate outcome can be obtained by taking into account, the result of the fuzzy logical units at multiple stages. The settings for the convolutional network and fuzzy logical units are integrated after end-to-end learning using training samples with physically labelled protein structures.

CNNs have gained popularity recently in the domains of bioinformatics and allied ones. Three principals have drawn researchers' attention to convolutional neural networks: shared weights, spatial subsampling, and local receptive fields. Convolutional neural networks perform several functions, including scale distortion and covariance shift, to varying degrees. Because of these characteristics, convolutional neural networks are frequently employed in research domains including segmentation and prediction. Genetic diversity is assumed to originate from proteins.

To forecast the secondary structure of proteins, features are extracted from sets of amino acids. CNN is unquestionably one of the greatest options for prediction when combined with its capacity to handle massive volumes of training samples. CNN shortens computation times while simultaneously capturing information from a large number of samples related to proteins. CNN Architecture for protein structure prediction shown in Figure 3.

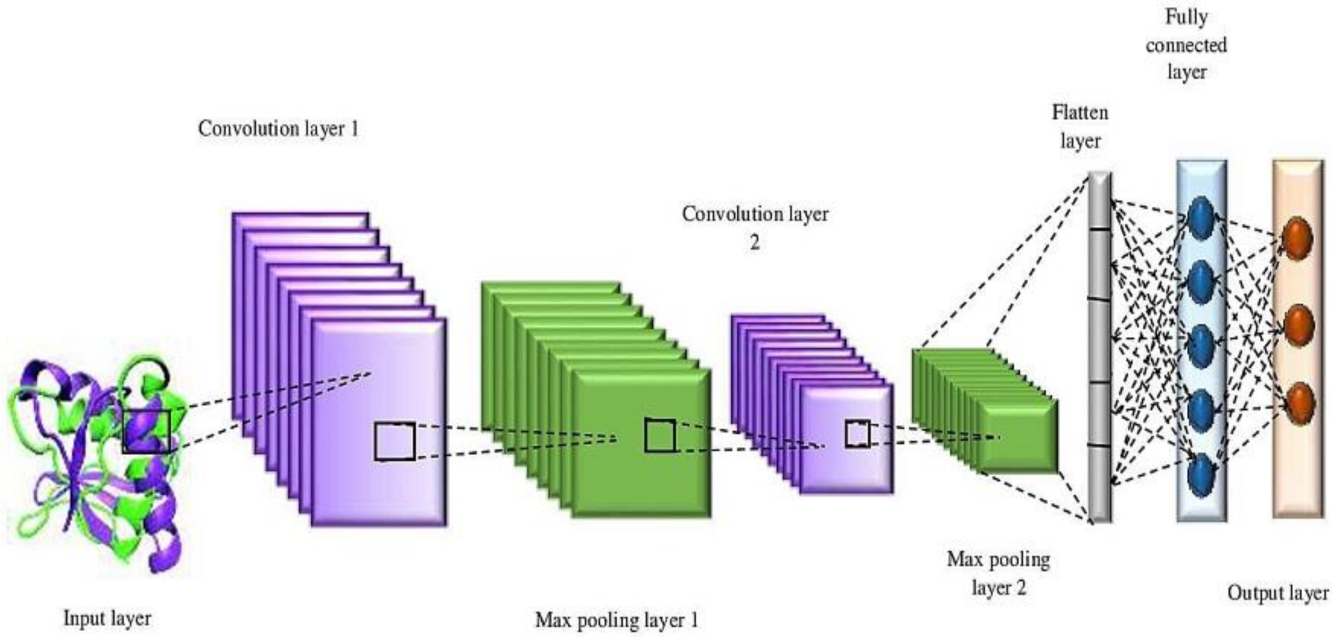


Figure 3. CNN Architecture for protein structure prediction

Step 2 involves an inspection of the input data by the convolution and max-pooling layers, which then produce the finest feature map F1, which is half the size of the image in both horizontal and vertical dimensions. Then, in step 2, a max-pooling layer and a residual block are used to extract all of the features at the appropriate resolution. Two max-pooling layer steps come after the remaining blocks. There is utilization of fully bonded layers. The mapping of representations between inputs and outputs is aided by the FC layer.

To build the finest feature for protein structure prediction, it selects partially the size in both the vertical and horizontal sides from the given input and processes it run over 2 convolutional layers first, then using a max-pooling level at a pace of two steps. The entire set of features in the matching resolution are then extracted using a remaining block is followed by a stride of two max-pooling layers. Fully connected layers are used. The FC layer assists in the mapping of representations between input and output.

Every membership function labels the feature points with a fuzzy linguistic term, and all of the Gaussian function membership are provided as

$$V_{x,y,k,c} = e^{-(\frac{k_{x,y,c}-\mu_{k,c}}{\sigma_{k,c}}).^{2}}, x = 1 \dots W, y = 1 \dots W, k = 1..M \dots \quad (1)$$

Where,

H, W The feature's height and width.

M Membership function applied for each function.

(x, y) coordinate feature point of $F_{x,y,c}$

$\mu_{k,c}$ and $\sigma_{k,c}$ The gaussian member function's mean and standard deviation

V $_{x, y, k, c}$ In channel c, the fuzzy logic feature's k-th output (x, y).

3.2. Optimization

For better accuracy in Fuzzy CNN, the Dingo Optimization Algorithm is used. This optimization uses a unique technique to tackle specific problems by altering the network's variables and derivatives to find the best solution. DOA's main concept is as follows: a rapid sequence to initialise the speed and position of the search agent, consequently increasing the rate of search agents, generating a huge numbers of search agents, and eventually finding the best agent. To achieve the best results, the algorithm employs 3 strategies to optimise the categorised output.

The first strategy is encircling, the search agents (Dingo) often seek nominal objectives while alone, but create groups while hunting substantial goals. Flow chart for Dingo Optimization Algorithm shown in Figure 4

$$\overrightarrow{m_\iota}(p+1) = \beta_1 \sum_{a=0}^{N} \frac{[\overrightarrow{\varphi_a(p)} - \overrightarrow{m_\iota}(p)]}{N} - \overrightarrow{m_*}(p) \tag{2}$$

Where, $\vec{m}(p+1)$ search agent's new position.

$\overrightarrow{m_*}(p)$ Best search agent.

$\overrightarrow{m_\iota}(p)$ Current search agent.

N Random Integer Numbers.

The second strategy is persecution, in which the search agent pursues the small prey separately until it is trapped.

$$\overrightarrow{m_\iota}(p+1) = \overrightarrow{m_\iota}(p) + \beta_1 * e^{\beta_2} * \left(\overrightarrow{m_s}(p) - \overrightarrow{m_\iota}(p)\right) \tag{3}$$

Where, $\vec{m}(p+1)$ dingo movement

s From 1 to the maximum size, a random number will be created.

The third strategy is scavenger, the search agent comes finds carrion to eat, they engage in scavenging activity while moving around their habitat at random, and then the fitness is determined.

$$\overrightarrow{m_\iota}(p+1) = \frac{1}{2}[e^{\beta_2} * \overrightarrow{m_s}(p) - (-1)^{\sigma} * \overrightarrow{m_\iota}(p)] \tag{4}$$

In addition, with 3 strategies, the chances of dingoes surviving are taken into account.

$$SR(i) = \frac{fitness() - fitness(i)}{fitness() - fitness(min)} \tag{5}$$

Where, *fitness (max)* is the greatest fitness

fitness (min) is the inferior fitness ratios

The low survival rate is given by,

$$\overrightarrow{m_\iota}(p) = \overrightarrow{m_*}(p) + \frac{1}{2}[\overrightarrow{m_s}(p) - (-1)^{\sigma} * \overrightarrow{m_t}(p)] \tag{6}$$

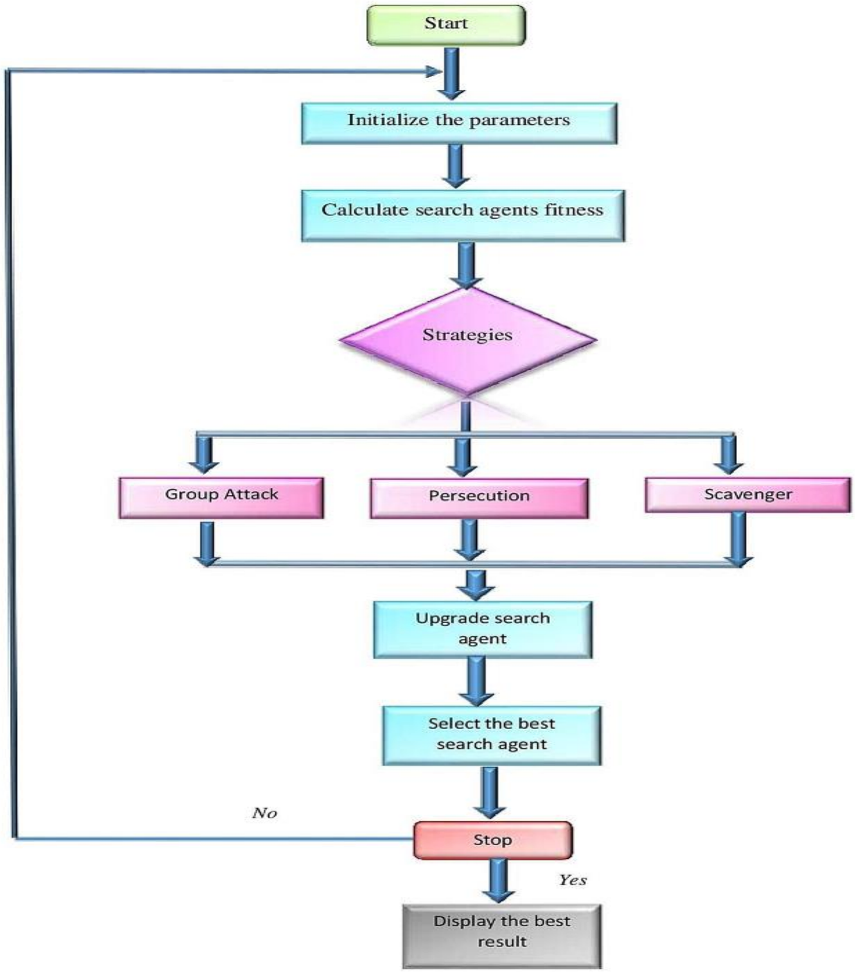


Figure 4. Flow chart for Dingo Optimization Algorithm

4. RESULTS AND DISCUSSIONS

The effectiveness of the Di-Fuzzy CNN technique is demonstrated in FPR (false positive rate), TPR (true positive rate), and is measured at the level of utilizing several datasets, including HSSP, PDB (Protein Data Bank), UGR14b, and DSSP (Secondary Structure Protein Data Bank). Protein structure prediction rate, PSPA (protein structure prediction accuracy), and PSPT (protein structure prediction time) are examined using tables and graphs in this section.

4.1. True Positive Rate experiment results

The amount of amino acid characteristics that are accurately chosen as related to the total amount of features are calculated in TPR. It is employed in the prediction of protein structure for feature selection. TPR is expressed as a % and can be computed using the equation below.

$$TPR = \frac{Amount\ of\ features\ selected}{Total\ amount\ of\ features} * 100 \qquad (7)$$

Table 1. The Result for TPR

Dataset	True positive Rate in percentage			
	BFO	PROTEUS	WPC - IRFC	Di-Fuzzy CNN
HSSP	82	83	92	94
PDB	84	87	95	97
UGR14b	80	82.7	90	93.4
DSSP	80.6	82	86	90

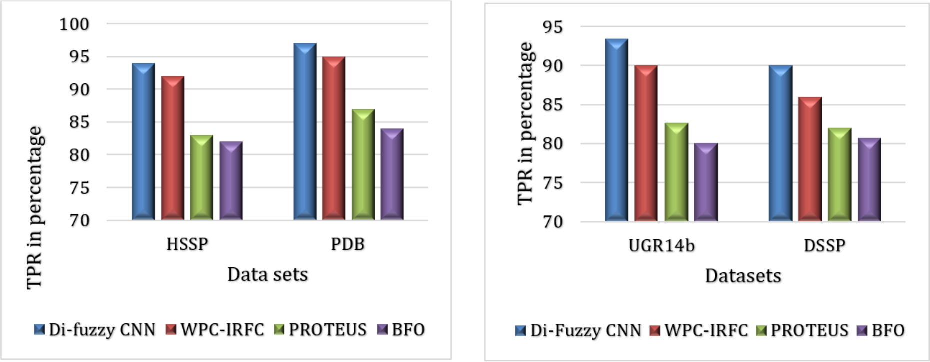


Figure 5. Graphical representation for TPR using different techniques and datasets

The outcomes of TPR in Di-Fuzzy CNN methods are compared to current methods such as WPC-IRFC, PROTEUS, and BFO for estimating protein structures from large protein datasets. When employing 100-550 amino acid characteristics from the PDB dataset in a study, the Di-Fuzzy CNN approach achieves a TPR of 97 percent, whereas previous methods reach only 95 percent, 87 percent, and 84 percent. Therefore, the proposed method outperforms existing methods. Graphical representation for TPR using different techniques and datasets shown in Figure 5. The Result for TPR shows in Table 1.

4.2. Rate of False Positives

The potential of mistakenly rejecting the null hypothesis for a test when making many comparisons is known as a false positive ratio. The amount of amino acid characteristics that are wrongly identified as significant to the total no. of features obtained as input is calculated. By using proposed technique FPR is low. Graphical representation for FPR using different techniques and datasets Figure 6. Comparison table for FPR Shows in Table 2

$$FPR = \frac{the\ no.\ of\ features\ that\ were\ incorrectly\ picked}{Total\ number\ of\ Features} * 100 \ (8)$$

Table 2. Comparison table for FPR

Dataset	False Positive Rate in percentage			
	BFO	PROTEUS	WPC-IRFC	Di-Fuzzy CNN
HSSP	17.4	17	7	6
PDB	16	13	5	3
UGR14b	15	12	6	5
DSSP	16.7	14	9	7

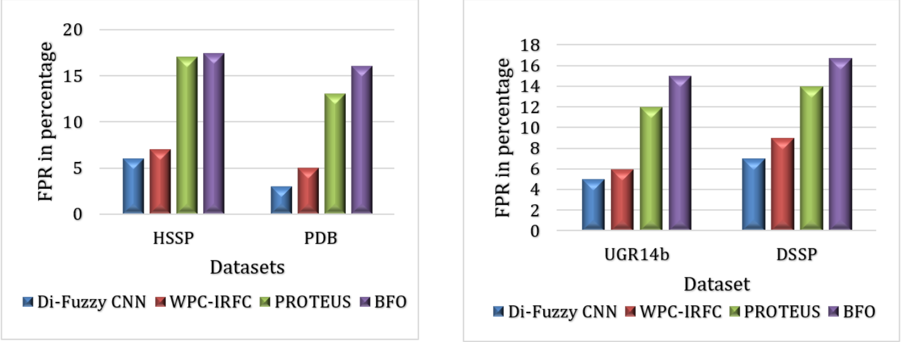


Figure 6. Graphical representation for FPR using different techniques and datasets

4.3. Structure of Proteins Experiment on prediction accuracy

It can be defined as the proportion of the total amount of protein structures that can be accurately predicted using particular features of amino acids.

$$PSPA = \frac{amount\ of\ properly\ predicted\ protein\ structure}{Total\ amount\ of\ protein\ structures} * 100 \qquad (9)$$

By using Di-Fuzzy CNN methods the results of PAPR is compared with different existing techniques such as WPC-IRFC, PROTEUS, and BFO for accuracy.

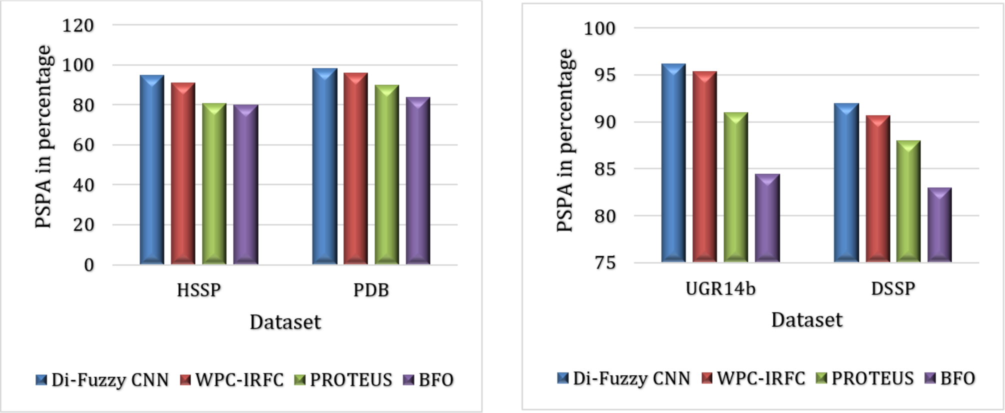


Figure 7. PSPAs graphical representation using different techniques and datasets

Table 3. Accuracy for predicting protein structure

Dataset	**Accuracy rate in%**			
	BFO	**PROTEUS**	**WPC-IRFC**	**Di-Fuzzy CNN**
HSSP	80	81	91	93
PDB	84	90	96	98
UGR14b	84.5	91	95.4	96.2
DSSP	83	88	90.7	92

PSPAs graphical representation using different techniques and datasets shows in Figure 7. Accuracy for predicting protein structure shows in Table 3.

By using PDB dataset the proposed technique achieves 98.2 percent, while previous methods reach only 96%, 90%, 84%. Therefore, the proposed method gives more accuracy compared to others.

4.4. Protein structure prediction time

PSPT is a metric that evaluates how long it takes to categorize the structure of protein from a large protein dataset file. It is measured in milliseconds (ms). Table for predicting time shown in Table 4.

$$PSPT = N * time(predicting\ the\ protein\ structure) \qquad (10)$$

Table 4. Table for predicting time

Dataset	**Time taken for predicting protein structure in ms**			
	BFO	**PROTEUS**	**WPC-IRFC**	**Di-Fuzzy CNN**
HSSP	28	27	16	14
PDB	25	26	13	11
UGR14b	24	22	19	16
DSSP	23	22.8	20	16.4

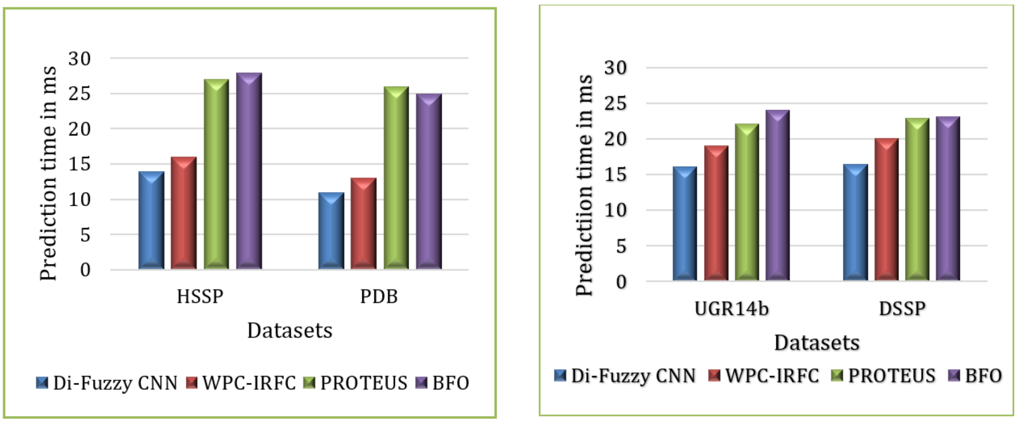


Figure 8. PSPTs graphical representation using different techniques and datasets

PSPTs graphical representation using different techniques and datasets Figure 8. The outcomes of PSPT in Di-Fuzzy CNN methods are compared to current methods such as WPC-IRFC, PROTEUS, and BFO for estimating protein structures from large protein datasets. When employing 100-550 amino acid characteristics from the PDB dataset in a study, the Di-Fuzzy CNN decrease the prediction time as 11 milliseconds, whereas previous methods reach only in 13 ms, 26 ms, 25 ms respectively. Therefore, the proposed method outperforms existing methods.

5. CONCLUSION

In this study, amino acid sequences were used to deduce the structure of proteins. In this article, we introduce a Di-Fizzy CNN that can accurately predict proteins from amino acids. It is proposed to learn high-level semantics through end-to-end supervised learning of convolutional structures in neural networks combined with fuzzy logic. Fuzzy allows convolutional neural networks to focus more on protein structure. The proposed method is tested on a large-scale protein dataset and compared using metrics such as TPR, FPR, PSPA, and PSPT. Compared with state-of-the-art features, Di-Fuzzy CNN result analysis achieves higher performance in terms of PSPA and PSPT, resulting in effective disease diagnosis. Future research should expand the dataset size and investigate the accuracy rates of deep learning models (e.g., RNNs, capsule networks) used to extract features from protein datasets. Explore spatial complexity and predict protein structure using large datasets.

CONFLICTS OF INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

FUNDING STATEMENT

Not applicable

ACKNOWLEDGEMENTS

The author would like to express his heartfelt gratitude to the supervisor for his guidance and unwavering support during this research for his guidance and support.

REFERENCES

[1] M.N. Mounika and M.N. Chava, “Protein Structure Prediction of Stroke Disease Using Artificial Neural Networks”, *Solid State Technology*, vol. 63, no. 2s, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[2] Y.S. Ding, T.L. Zhang and K.C. Chou, “Prediction of protein structure classes with pseudo amino acid composition and fuzzy support vector machine network”, *Protein and peptide letters*, vol. 14, no. 8, pp. 811-815, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[3] J. Xu, M. Mcpartlon and J. Li, “Improved protein structure prediction by deep learning irrespective of co-evolution information”, *Nature Machine Intelligence*, pp. 1-9, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[4] H.K. Phan and T.H. Dang, “Protein structure prediction using Deep Learning”, *VNU University of Engineering and Technology*, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[5] A.K. Mandle, P. Jain and S.K. Shrivastava, “Protein structure prediction using support vector machine”, *International Journal on Soft Computing*, vol. 3, no. 1, pp. 67, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[6] S.C. Pakhrin, B. Shrestha, B. Adhikari and D.B. Kc, “Deep Learning-Based Advances in Protein Structure Prediction”, *International Journal of Molecular Sciences*, vol. 22, no. 11, p.5553, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[7] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko and A. Bridgland, “Highly accurate protein structure prediction with AlphaFold”, *Nature*, vol. 596, no. 7873, pp. 583-589, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[8] S. Montgomerie, S. Sundararaj, W.J. Gallin and D.S. Wishart, “Improving the accuracy of protein secondary structure prediction using structural alignment”, *BMC bioinformatics*, vol. 7, no. 1, pp. 1-13, 2006. [CrossRef] [Google Scholar] [Publisher Link]

[9] S.J. Giri, P. Dutta, P. Halani and S. Saha, “MultiPredGO: Deep multi-modal protein function prediction by amalgamating protein structure, sequence, and interaction information”, *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 5, pp. 1832-1838, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[10] S. Zhou, H. Zou, C. Liu, M. Zang and T. Liu, “Combining deep neural networks for protein secondary structure prediction”, *IEEE Access*, vol. 8, pp. 84362-84370, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[11] Z. Bingzhen, Q. Xiaoming, Y. Hemeng and Z. Zhubo, “A Random Forest Classification Model for Transmission Line Image Processing”, *In 2020 15th International Conference on Computer Science & Education (ICCSE) IEEE*, pp. 613-617. 2020. [CrossRef] [Google Scholar] [Publisher Link]

[12] S. Akter and L. Holder, “Improving IoT Predictions through the Identification of Graphical Features”, *Sensors*, vol. 19, no. 15, p.3250, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[13] M. Gao, H. Zhou and J. Skolnick, “DESTINI: A deep-learning approach to contact-driven protein structure prediction”, *Scientific reports*, vol. 9, no. 1, pp. 1-13, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[14] K.S. Zamil and J. Rahman, “Prediction of Protein-Protein Interaction from Amino Acid Sequence Using Ensemble Classifier”, *In 2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2) IEEE*, pp. 1-4. 2018. [CrossRef] [Google Scholar] [Publisher Link]

[15] B.Ç. Yavuz, N. Yurtay and O. Ozkan, “Prediction of protein secondary structure with clonal selection algorithm and multilayer perceptron”, *IEEE Access*, vol. 6, pp.45256-45261. , 2018 [CrossRef] [Google Scholar] [Publisher Link]

[16] S. Xie, Z. Li and H. Hu, “Protein secondary structure prediction based on the fuzzy support vector machine with the hyperplane optimization”, *Gene*, vol. 642, pp. 74-83, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[17] Y. Wang, H. Mao and Z. Yi, “Protein secondary structure prediction by using deep learning method”, *Knowledge-Based Systems*, vol. 118, pp.115-123, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[18] Y. Liu, J. Cheng, Y. Ma, and Y. Chen, “Protein secondary structure prediction based on two dimensional deep convolutional neural networks”, *In 2017 3rd IEEE International Conference on Computer and Communications (ICCC) IEEE*, pp. 1995-1999, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[19] A.K. Pani, M. Manohar and R. Kumar, “An efficient algorithmic technique for feature selection in IoT based intrusion detection system”, *Indian Journal of Science and Technology*, vol. 14, no. 1, pp. 76-85, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[20] H. Peraza-Vázquez, A.F. Peña-Delgado, G. Echavarría-Castillo, A.B. Morales-Cepeda, J. Velasco-Álvarez, and F. Ruiz-Perez, “A bio-inspired method for engineering design

optimization inspired by dingoes hunting strategies", *Mathematical Problems in Engineering*, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[21] C. Guan, S. Wang and A.W.C. Liew, 2019, "Lip image segmentation based on a fuzzy convolutional neural network", *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 7, pp. 1242-1251. [CrossRef] [Google Scholar] [Publisher Link]

[22] D.A. Pelta and N. Krasnogor, "Multimeme algorithms using fuzzy logic-based memes for protein structure prediction", *In Recent advances in memetic algorithms* Springer, Berlin, Heidelberg, pp. 49-64, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[23] B. Kuhlman and P. Bradley, "Advances in protein structure prediction and design", *Nature Reviews Molecular Cell Biology*, vol. 20, no. 11, pp. 681-697, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[24] D.E. Kim, D. Chivian and D. Baker, "Protein structure prediction and analysis using the Robetta server", *Nucleic acids research*, vol. 32, no. suppl_2, pp. W526-W531, 2004. [CrossRef] [Google Scholar] [Publisher Link]

AUTHORS

P. G. Sreelekshmi currently working in University Institute of Technology Malayinkeezhu under University of Kerala as Lecturer. Prior to her recent appointment she was worked at Sivaji College of Engineering as Assistant Professor. She completed her ME in 2014 from Anna University Chennai.

S.C. Ramesh completed a bachelor of engineering in electronics and communication engineering, master of engineering in avionics and doctor of philosophy in Anna University Chennai, India. He is working as associate professor in PSN College of engineering and technology, Tirunelveli, India

Arrived: 24.11.2023
Accepted: 23.12.2023

International Journal of Data Science and Artificial Intelligence (IJDSAI)
Volume 01, Issue 02, November – December (2023)

RESEARCH ARTICLE

IOT-ENABLED PROTEIN STRUCTURE CLASSIFICATION VIA CSA-PSO BASED CD4.5 CLASSIFIER

T. Maris Murugan[1, *] and A Jeyam[2]

[1] Erode Sengunthar Engineering College, Perundurai, Erode, Thuduppathi, Tamil Nadu 638057 India.
[2] Department of Electrical and Electronics Engineering, Erode Sengunthar Engineering College, Perudurai, Erode, 638057 India

*Corresponding e-mail: marismurugan428@outlook.com

Abstract – Data mining is a technique for obtaining useful information from vast amounts of information. Big data refers to large amounts of complicated information that is processed, particularly in relation to biological processes. The investigation of protein structures has recently received a lot of attention from structural biologists. The majority of recent research projects have tried to improve protein structure identification in huge data. Feature selection-based protein structure identification in large data analysis, on the other hand, takes a long time. A hybrid crow search algorithm and particle swarm optimization (CSA-PSO) based CD4.5 (CP-CD) approach has been developed to increase Protein Structure Identification accuracy with less amount of time. First samples from the patients are given to IOT-enabled microscope and the details will be stored in big data and then the process will be divided into two steps. At first, feature selection is done using CSA-PSO algorithm, and the classification is done using CD4.5 classifier. This aids in identifying the protein structure and accurately diagnosing the condition, as well as lowering the false positive rate.

Keywords – *Protein structure classification, feature selection, Big data analysis, CD4.5 classifier, IOT-enabled microscope.*

1. INTRODUCTION

Proteins are important in biological activities and are collected of amino acids connected together by peptide bonds. The three-dimensional structure of atoms in a protein molecule is known as protein structure. Protein structures are analyzed using nuclear magnetic resonance (NMR) spectroscopy or X-ray crystallography. Protein sequence is a method of determining a protein's amino acid sequence or structure. The sequence and three-dimensional (3D) structure of a protein influence its function. At an unprecedented rate, large-scale genome sequencing efforts are supplying researchers with millions of protein sequences from numerous species. [1]. Enzymatic catalysis, transferring ions and chemicals from one organ to other, nutrition, the contractile system of muscles, tendons, cartilage, antibodies, and modulating cellular and physical processes are all roles performed by proteins [2]. Structure of protein shown in Figure 1.

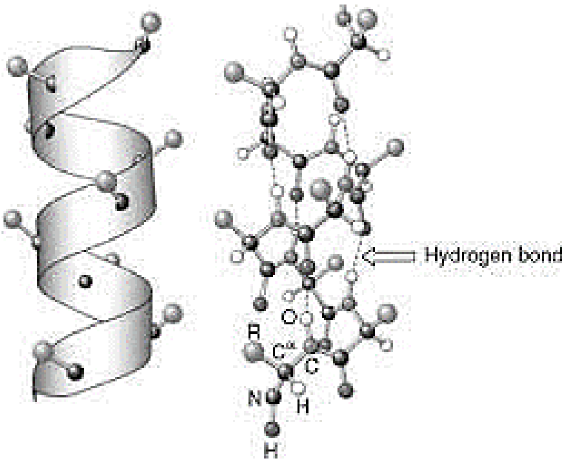


Figure 1. Structure of protein

The primary structure in a polypeptide chain is the arrangement of amino acids. The regular, repeating spatial groupings of nearby amino acid residues in a polypeptide chain are referred to as secondary structure. The amide hydrogens and carbonyl oxygens are tightly bonded together to form the peptide backbone. Helixes and structures are the most common types of secondary structures [2]. In medical diagnostics, identifying the protein structure is essential in determining the disorder. Feature selection is the process of choosing the most important inputs to process and analyze, or reducing the total amount of inputs. Increasing the predictive model's performance while reducing the modeling cost is the main goal of feature selection [3].

The Internet of Things, or IoT, is a network of individually addressable physical items that can interact and communicate with each other through the Internet. These objects have varied degrees of processing, sensing, and actuation capabilities. Therefore, the main objective of the IoT is to allow objects to connect with people and other objects at any time and from any place by using any network, technique, or service [4]. The simplest method to assess an algorithm is to check at all subsets of potential functions and see what decreases the proportion of errors. It is a systematic exploration for space that benefits everyone in the calculation except the smallest feature set. Wrappers, filters, and embedding techniques are three different types of feature selection algorithms with different test evaluations. The term feature selection is often used in data mining to limit inputs to a manageable size for analysis and processing, with an emphasis on discovering relevant content without compromising the classification algorithm's accuracy [5].

Bioinformatics and theoretical chemistry work together in medical applications to analyse protein structure big data. The majority of recent research projects have tried to improve protein structure recognition in large datasets. However, in big data analysis, feature selection-based protein structure identification does not save time. As a result, feature selection is required to establish the structure of the protein. And for category label, the key characteristic has a powerful meaning and significance. The duplicated functions, on the other hand, not only affect the algorithm's classification performance but also contribute to the processing expenses. With the feature selection process, which selects the best feature subset of the original feature domain, it is critical to reduce unwanted and redundant features.

The abovementioned defect causes lots of new problems, including a failure to choose relevant features, lower classification accuracy and longer identification times, a large false positive rate, and so on. To deal with such problems, hybrid CSA-PSO algorithm based CD4.5 classification (CP-CD) method has been presented. In this method patient's samples will be acquired and then given to an IOT-enabled microscope. The findings will be kept as large data, and the procedure will be divided into two steps. In the first stage, a hybridised crow search and particle swarm optimization technique is used to identify features. The classification process is then accomplished in the second stage.

The other sections of the document are arranged under the respective topics. Section II provides a description of the literature review. Section III provides a description of the suggested method. Section IV provides a description of the findings and discussion. Section V provides a description of the conclusion.

2. LITERATURE REVIEW

Protein sequence classification is a crucial process for identifying the disease in humans. There is lot of research on this protein structure classification, among these a few are discussed here.

In 2021, Sequeira, et al, [6] proposed a method called ProPythia, a generic and modular Python program that permits users to quickly apply ML and DL algorithms to a variety of protein sequence prediction and classification challenges. It makes it easier to implement, compare, and validate the main errands in ML or DL pipelines, such as modules for reading and altering series, calculating protein features, preprocessing datasets, and dimensionality drop, feature selection gathering and diverse scrutiny, and training and optimizing ML/DL models and using them to create forecasts. They also compare the presentation of the various forms in four distinct protein categorization problems.

In 2020, Kalaiselvi, and Thangamani, [7] have proposed a Weighted Pearson Correlation based Improved Random Forest Classification (WPC-IRFC) Technique. The WPC-IRFC method was created with the goal of improving protein structure prediction accuracy while saving time. WPC-IRFC Technique achieves 7% FPR in an experimental assessment utilising 50-500 amino acid characteristics from VariBench DS, whereas previous techniques achieve 22 percent, 20 percent, 17 percent, and 14 percent. As a result, the FPR of the WPC-IRFC method is smaller than other approaches.

In 2020, Ge et al., [8] proposed a step-by-step classification approach on the basis of double-layer SVM model to calculate the proteins' secondary structure. This approach is evaluated using a frequently used dataset, the 25PDB dataset, which has a sequence similarity of less than 40%. Despite the fact that these two models' accuracy is somewhat reduced, the correctness of the α+β and α/β classes is upto 85.09 percent and 78.64 percent, respectively, and the correctness of the α+β class is greater than existing techniques. The findings reveal that this technique performs well, and the correctness of α+β class proteins is greatly enhanced by assuring the correctness of the other three structural classes of proteins.

In 2017, Shu, and Yong, [9] describes a method for classifying protein secondary structures on the basis of protein "signal-plotting" and digital signal processing using the Fourier methodology. It has been shown that a larger variety of protein secondary structures may be categorized using these indices, which are the hydrophobicity rate and the dominant frequency. Finally, it is hoped the discovery will usher in a whole new era of protein secondary structure analysis, as well as DNA and protein sequence analysis. The findings indicate that these newly proposed indices can classify a greater variety of protein secondary structures.

In 2017, Najibi, et al., [10] developed a nonparametric approach for estimating numerous bivariate density functions for a group of populations with protein backbone angles. The suggested approach would be more effective than previous methods. The adaptive basis expansion coefficients for the fitted densities give a low-dimensional depiction of the densities which may be used for conception, grouping, and identification. The proposed method takes a novel and innovative approach to two important and challenging problems in protein structure research: structure-based protein classification and angular-sampling-based protein loop structure prediction.

In 2019, Ahmad, and Hayat, [11] proposed a revolutionary high-throughput computational methodology for accurately identifying subGolgi proteins. The publicly accessible benchmark dataset is very unbalanced, with trans-Golgi sequences accounting for 72 percent of the entire dataset. The high-rank features are chosen using a maximum vote technique, which reduces the feature space by 85 percent. The results show that using a KNN classifier in conjunction with a hybrid feature space yielded good results. It has a jackknife cross-validation accuracy of 98 percent, individual data accuracy of 94 percent, and a 10-fold cross-validation accuracy of 96 percent.

In 2020 Ahmad, et al., [12] have suggested a method that employs numerical descriptors based on sequences and evolution, primary protein sequences are constructed in this work. While evolutionary characteristics are gathered utilising a bigram scoring matrix customized to positions, sequential information is extracted employing K-space amino acid pair (KSAAP) and dipeptide composition. SVM with ideal features had a correctness of 97.54% for the training dataset and 93.71% for the independent dataset, respectively. Their suggested model was shown to outperform and provide the best results among the current computational models.

In 2020, Mirceva, et al., [13] presented a method for categorising protein shapes in this work. The results revealed that filtering 20 or 30 of the most significant attributes only slightly reduces performance in general. Only the C4.5 classifier is exempt from this, but this is due to the nature of this classifier, which picks the features with the best information gain throughout the model induction phase. The earlier results concerning the minor drop in accuracy by decreasing to 20 and 30 features are crucial since it suggests that the time required for training and testing the models might be cut in half with feature selection while still maintaining high accuracy.

In 2019, Mirceva, et al., [14] suggest a method for categorising protein structures in this work. They create models by combining several categorization algorithms. The proposed technique is thoroughly examined, as well as the advantages of using feature selection. The results demonstrate that feature selection produces superior outcomes in virtually all circumstances than when no feature selection is used. The investigation's overall conclusion was that most of the ways in the analysis perform better than the protein voxel-based descriptor, even though it beats several of the strategies.

In 2020, Ghosh, et al., [15] suggested a ML–based approach for classifying secondary structure of proteins into four categories: all-α, all-β, $\alpha+\beta$, and/. On the four standard datasets 640, 1189, 25pdb, and fc699, the overall accuracies achieved using the proposed model are 86.89 percent, 92.93 percent, 91.38 percent, and 94.87 percent, respectively. In this comparison, the suggested model outperforms certain state-of-the-art approaches.

3. PROPOSED METHOD

Protein structure identification is crucial for disease detection in big data analysis. Several data mining techniques, such as gene structure, DNA sequences, and protein sequences, have been developed in the disease diagnostic area. To eliminate the problem of mystery cases and prediction analysis during illness diagnosis, the suggested approach, protein sequences identification, is used. The suggested method efficiently identifies protein structures for brain tumour diagnosis.

Protein structure is the three-dimensional configuration of atoms within an amino acid chain molecule. Peptide bonds are formed by the condensation of amino acids to create protein structures. The terminus of a peptide or protein sequence with a free carboxyl group is called the carboxy-terminus, or C-terminus. The termini of a sequence with a free -amino group are denoted by the terms amino-terminus and N-terminus. Proteins are composed of twenty different compounds called amino acids. The citric acid cycle, Glycolysis, and the pentose phosphate pathway all offer intermediaries that are used to make amino acids. The 20 amino acids are made up of both essential and non-essential amino acids. Nine amino acids are essential, whereas the remaining nine are non-essential. The genetic code determines the amino acid sequence in a protein as well as its function. 20 different types of Amino acids shown in Table 1.

Table 1. 20 different types of Amino acids

Glycine	Gly	G	Tyrosine	Try	Y
Alanine	Ala	A	Methionine	Mer	M
Serine	Ser	S	Tryptophan	Trp	T
Threonine	Thr	T	Asparagine	Asn	A
cysteine	Cys	C	Glutamine	Gln	G
Valine	Val	V	Histidine	His	H
Isoleucine	Ile	I	Aspartic Acid	Asp	A
leucine	Leu	L	Glutamic Acid	Glu	G
Proline	Pro	P	Lysine	Lys	L
Phenylalanine	Phe	P	Arginine	Arg	A

Protein materials are composed up of a precise order of amino acids. The amino acid sequence is indicated on these strings. As a result, the erection of a protein explains the specific classification in which amino acids are connected together by peptide bonds to create a protein.

Fig 2 represents the flow of proposed methodology. In bioinformatics, protein structure identification is a critical step. Many factors contained in the training data set may increase the risk of correctly identifying the protein structure in real-world applications. As a result, for protein structure big data analysis to diagnose brain tumour illness and reduce the risk in protein structure identification, feature selection and classification are necessary. Attribute selection from a big dataset is also known as feature selection. The protein structure is then identified using the classification technique.

In the proposed technique, the samples of the patients are collected and tested using an IOT enabled microscope and the details will be automatically send to the big data cloud and also it informs the hospital so that the information can be accessed remotely. The proposed CP-CD technique consists of two processing steps: feature selection and classification, which allow for quick protein structure identification. For feature selection in the initial stage, a hybrid CSA-PSO approach is applied. The CD4.5 classifier is used to classify the selected features in the second stage. This aids in improving the efficiency of bioinformatics data processing while also saving time.

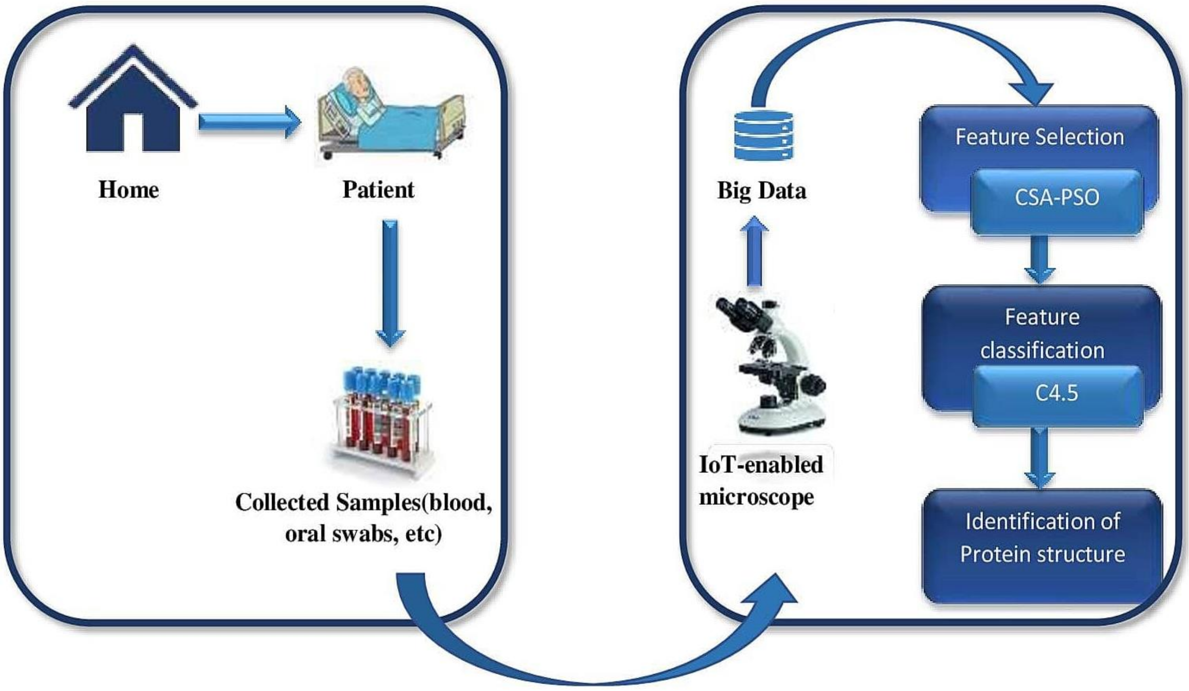


Figure 2. Schematic representation for proposed framework

3.1. Feature selection using CSA-PSO algorithm

The hybrid Crow Search Algorithm and Particle Swarm Optimization (CSA-PSO) algorithm. This algorithm hybrids the properties of crow search algorithm and particle swarm optimization algorithm which will give better results for feature selection from the large dataset.

a) Crow search algorithm

Crows were used as inspiration for the crow search algorithm because of their habit of keeping food in a secret area and recovering it after several months. Crows, like other social animals, may participate in thievery at some point by carefully studying other Crows' food concealing locations and then robbering their food. When a crow doubt that other is following him, he flees to a position faraway from where the food is hidden to deceive the thief. The CSA technique is linked with chaotic series in this method and it is represented as,

$$s_x^{(r+1)} = \quad \{s_x^{(r)} + J_y * Ht_y^{(r)} * \left(n_y^r - s_x^{(r)}\right) \quad J_i \geq BQ_t^r \quad choose\ a\ random\ position \quad otherwise \tag{1}$$

b) Particle Swarm Optimization algorithm

The attribute choices on the basis of social characteristics related with bird flocking to resolve optimization issues attract a lot of academic attention in Particle Swarm Optimization (PSO). PSO, which is a type of swarm intelligence optimization, has been shown to be lesser computationally expensive and to settle more quickly. Every solution in PSO may be seen as a swarm of particles, each with its own velocity and position.

c) Opposition Based Learning:

The OBL approach searches in both directions in the search space. One of these two pathways contains the initial answer, while the other indicates the opposite direction. The opposite location in M-Dimensional space with s (s_1……s_m) and $s_1 \epsilon [\delta, \gamma]$, x=1,2,3….M is calculated in the below equation

$$s_x^{\overline{r}{opp}} = \delta_x + \gamma_x - s_x^r \tag{2}$$

d) CSA-PSO algorithm:

The concepts of binary CSA and binary PSO algorithms have been mixed, resulting in a technique called CSA-PSO that benefits from their inclusion. For example, in CSA-PSO approach, only aiming particular crows with better foods improves the execution of randomly following each crow. The Opposition Based Learning approach is then used to create the crows' opposite positions, which are subsequently utilized to upgrade the post in the PSO. This is achieved so that both methods can examine the exploration space in turn, without being impacted by the results of the other.

Fig 3 represents the feature selection process using this CSA-PCO method. The first step is preprocessing, that is to get the details we needed from the big dataset are to preprocessed for further process. In CP-CD technique, CSA-PSO is the hybrid technique of crow search and particle swarm optimization method.

Logistics map:

$$s_x^{r+1} = bs_x^r(1 - s_x^r) \quad b = 0.4\ and\ s_1 = 0.7$$

Exponential map: $s_x^{r+1} = s_x^r e^{2(1-s_x^r)}$ s_1=0.7

$$J_{r+1} = v + s_x^{r+1},$$

Where v is the energetic parameter that controls s_x^ractivity. When v steps up, s_x^r undergoes further bifurcations, eventually resulting in pandemonium. The current situation would change and the Crow would move to the right answer if a predetermined random number was less than this threshold value.

$$U_{shape} = \left|\frac{2}{\pi}\ arctan(\frac{\pi}{2} s_x^r)\right| \quad (3)$$

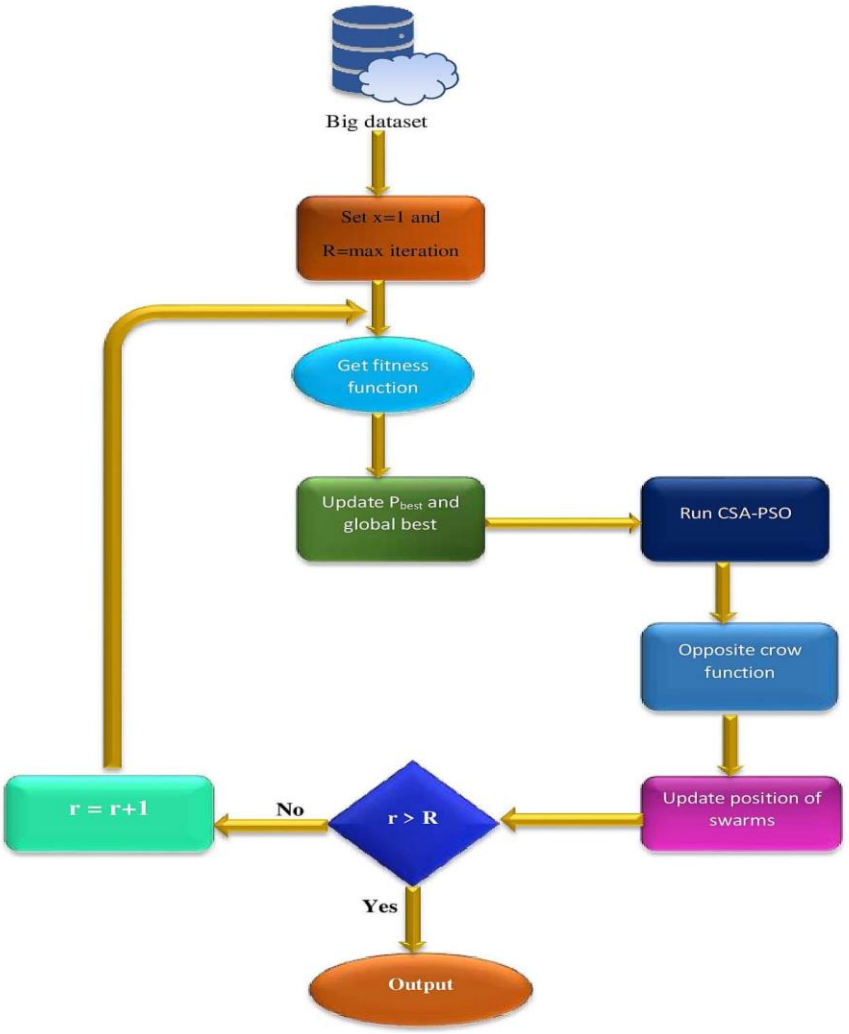


Figure 3. Flow diagram that represents feature selection using CSA-PCO

e) Fitness function

Algorithm 1: CSA-PSO based feature selection algorithm

Algorithm for CSA-PSO

```
Begin
Initialize x=0
Wgt= $wgt_{max}$-iteration $(\frac{wgt_{max}-wgt_{min}}{max-iteration})$
Evaluation of fitness value
$P_{best}$, $g_{best}$ values are set
Run CSA-PSO with $S_x^r$ population
Reversely alter the position that the CSA-PSO returned.
for (x=1; x≤ $D; x++$) do
    if$(f(S_x^r) \geq crowmin) do$
        $\overline{s_x^r}^{opp}$=$\delta_x + \gamma_x - s_x^r$
    end if
end for
upgrade swarm position
for v=1 to QQ
    $U_x^{r+1} = wgtu_s^r + J_1 n_{x1}\left(P_{best}{}_x^r - S_x^r\right) + J_2 n_{x2}(G_{best}{}_x^r - S_x^r)$
end
for v=1 to QQ
    for y=1 to N
        if (u(x,y) > $U_{max}$)
            u(x,y) = $U_{max}$
        end
        if(u(x,y)< -$U_{max}$)
            u(x,y)=-$U_{max}$
        end
        q=$\frac{1}{1+e^{-u(x,y)}}$
        If (nand < q)
            $S_{x,y}^{r+1} = 1$
        else
            $S_{x,y}^{r+1} = 0$
        end
    end
end
r=r+1
Produce best results
```

The equation defines the fitness function for finding results to attain a balance between the two objectives.

$$fitness = \delta\Delta_D(N) + \gamma\frac{|Z|}{|R|} \quad (4)$$

$\Delta_D(N)$ represents the error rate of classifier, $|Z|$ represents the subset's size which the method chooses and $|R|$ represents the absolute number of features in the existing dataset. δ is a parameter ∈ [0,1] associating to weight of error rate for classification. $\gamma = 1 - \delta$ represents the importance of decrease in feature.

The given algorithm is utilized to choose pertinent characteristics for categorization from a large dataset. To construct a protein structure, the properties that are most closely associated to the amino acid are chosen. This contributes to a higher true positive rate.

f) CD4.5 machine learning classifier

The classification is done using the c4.5 machine learning classifier after the relevant features from the huge dataset have been selected. The C4.5 technique is employed in data mining as a Decision Tree Classifier, which may be used to decide on the basis of a sample of data. Classification using CD4.5 classifier shown in Figure 4.

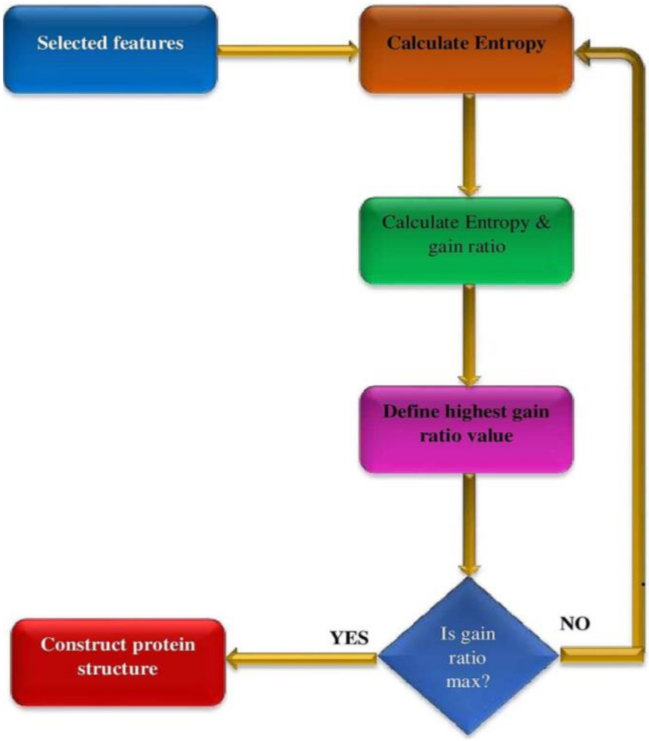

Selected features
Calculate Entropy
Calculate Entropy & gain ratio
Define highest gain ratio value
Is gain ratio max?
YES
NO
Construct protein structure

Figure 4. Classification using CD4.5 classifier

Ross Quinlan created the C4.5 algorithm, which is used to build decision trees. C4.5 extends the ID3 methodology. C4.5 is referred to as a statistical classifier since it generates decision trees that may be used to categorize data. Similar to ID3, C4.5 uses the concept of information entropy to build decision trees from a collection of training data.

g) Information Entropy

Information gain is the reduction in entropy produced by modifying a dataset, and it is commonly used in the training of decision trees. To measure information gain, the entropy of a dataset before and after a mutation is employed.

$$H(z) = -\sum_{x=1}^{m} p(y_j)p(y_j) \quad (5)$$

If a distinct classification for the resultant feature can be established for each of the feature values, the information gain is equivalent to the total entropy for that attribute. The relative entropies removed from the overall entropy are 0 in this scenario. The training dataset is a set D=d_1,$d_{2,...}$ of already classified samples. Each sample displaystyle d_i is made up of a m-dimensional vector display style ($y_{1,j}$,$y_{2,j}$,….$y_{m,j}$), where the y_i reflect the sample's attribute values or features, and the class in which y_j falls. C4.5 picks the properties of data which efficiently separates the sample set into subsets overloaded in one class or the other at every node. The normalized information gain is employed as a dividing criterion. The characteristic with the largest standardized information gain is selected for selections. The C4.5 method then iterate through the subdivided subsets. The CD4.5 decision tree is constructed as follows, using entropy and information gain calculations. Decision tree structure for CD 4.5 classifier shown in Figure 5.

Algorithm for CD4.5 classifier

Input: Training dataset D, features selected

Output: Classification of protein structure

Step:1 Examine the above-mentioned base cases.

Step:2 Find the standardized information gain ratio from dividing on a for each attribute d.

Step:3 Assume that d_{best} has the maximum normalized information gain.

Step:4 Make a decision node that splits based on the value of d_{best}.

Step:5 Recur on the subsets formed by separating on a best and append them as children to node.

End

The above diagram represents the decision tree structure for CD 4.5 classifier. A, B, C are the features. There are root nodes and leaf nodes in a typical decision tree. Features are used to represent the nodes. A subset of characteristics is represented by each node's decision. With a class label, the leaf node is also known as the tree's terminal node. As a selection, the feature with the largest information gain is picked. Every route from the root node to the leaf node in the decision tree creates a categorization rule. A decision tree is a type of recursive classification classifier. Every leaf node in the diagram represents a categorization judgement of characteristics to construct a protein structure. This reduces the number of false positives.

There are a few fundamental uses for this technique.

- All of the lists' collections belong to the same category. All that happens is that a leaf node telling the user to choose that class is added to the decision tree.
- None of the characteristics yield any information. In this case, C4.5 builds a decision node based on the anticipated rate.

- A class instance that had never been observed before occurred. C4.5 builds a decision node using the anticipated rate.

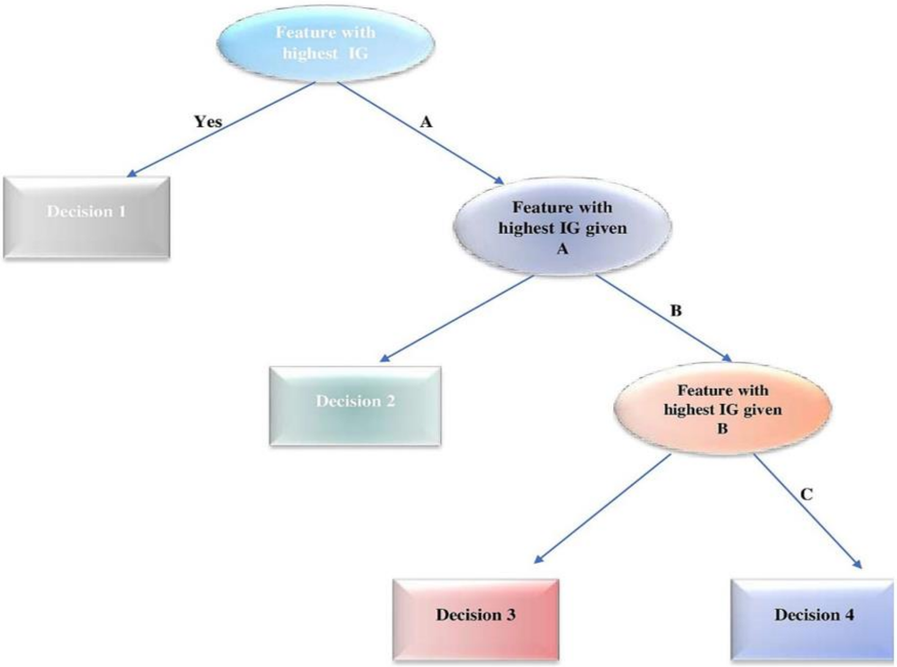


Figure 5. decision tree structure for CD 4.5 classifier

The algorithmic description of a CD4.5 decision tree method to pick the attributes of amino acid sequence to determine the protein structure is shown in algorithm 2. To categorise the best and greatest gain splitting features, entropy value and information gain are assessed for each picked feature from the huge dataset. After determining the best features, every node denotes a feature test, and every leaf node denotes a class label, the amino acid sequence may be identified. To build a protein structure, certain amino acid sequence characteristics are chosen. These speeds up the process of identifying the protein structure.

4. RESULTS AND DISCUSSION

For determining the protein structure, a CSA-PSO and CD4.5 classifier (CP-CD) approach is tested utilising the JAVA language and the Weka tool. The proposed CSA-PSO based CD 4.5 classification (CP-CD) technique is compared to 4 existing models: the Logistic Regression based Iterative Dichotomiser 3 classification (LR-ID3) technique, the Hydrogen-deuterium Exchange measured by Nuclear Magnetic Resonance (HDX-NMR) technique, the TAndem PrOtein detector (TAPO) method, and the Protein secondary structure prediction (PSSP) technique. Five datasets (i.e., Protein Data Bank (PDB), ProteinNet, PROSITE, sidechainNet, and pfam dataset) are utilized to effectively identify the protein structure to illustrate the benefit of the proposed CP-CD approach. It contains data on 3D protein shapes, nucleic acids, and complex assemblies, all of which are used to determine the importance of proteins in terms of health and illness.

The purpose of the PDB database is to classify and describe protein structures while also giving biological data. SEQRES (i.e. Residues in the Sequence) entries in the PDB database include the sequences of the three peptide chains A, B, and C. It is used in a variety of fields such as molecular biology, structural biology, and computational biology.

ProteinNet is a standardised data collection for protein structure machine learning ProteinNet relies on the biennial CASP evaluations, which include making blind predictions of freshly resolved but publicly accessible protein structures, to provide test sets that push the boundaries of computational methods.

PROSITE was the world's first secondary database. Most protein families have certain highly conserved motifs that may be decoded to determine diverse biological activities. When a new sequence is found, we may quickly determine the protein family by utilising a database tool like this. PROSITE is essential in this regard. A regular expression is used to encode motifs in PROSITE (called patterns).

Sidechain Net is an extension of ProteinNet1 dataset for protein structure prediction. In particular, Sidechain Net replaces the protein backbone with measurements for protein angles and coordinates that characterize the whole, all-atom protein structure (backbone and sidechain, excluding hydrogens).

The Pfam database's major goal is to give a precise and detailed identification of protein sequences. The goal of building the database is to increase genome annotation efficiency.

The true positive rate, protein structure identification accuracy, false positive rate, and protein structure identification time, recall, precision, and F-measure are measure against existing approaches to calculate the performance of CP-CD technology.

a) True Positive rate

It is expressed as a percentage of the amount of correctly picked features when measure against the total number of characteristics in the big data for protein structure identification (%).

$$True\ Positive\ Rate(TPR) = \frac{Number\ of\ features\ selected\ correctly}{Total\ number\ of\ features} \times 100 \qquad (6)$$

The procedure is believed to be more efficient if the true positive rate is much higher. Table 2 shows the experimental findings for true positive rate based on the number of characteristics.

Table 2. Comparison of true positive rate

Dataset	True Positive Rate				
	PSSP	HDX-NMR	TAPO	LR-ID3C	CP-CD
PDB	72	78	80	85	90
ProteinNet	77	80	83	86	92
PROSITE	82	84	87	88	93
SidechainNet	84	85	88	92	94
pfam	85	88	92	94	97

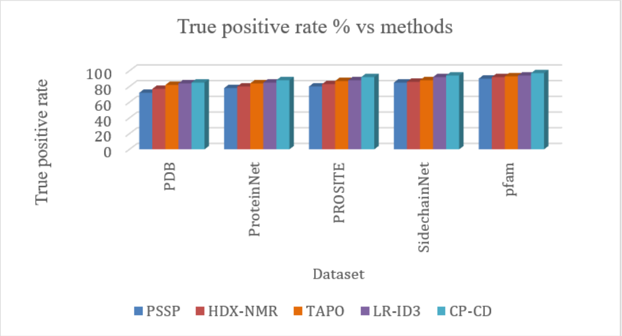


Figure 6. Comparison of true positive rate of CP-CD technique with other existing methods

Figure 6 represents that the suggested CP-CD methodology achieves a true positive rate of 97 percent for 500 protein characteristics, whereas current techniques such as LR-ID3C, TAPO, HDX-NMR and PSSP provide true positive rates of 94 percent, 92 percent, 88 percent, and 85 percent, respectively, as shown in Table 2.

b) Protein structure identification accuracy

The relation of number of features with least entropy and highest information gain employed to create the decision tree to the number of features in the database is described as protein structure identification accuracy in the CP-CD approach. The formula for determining the accuracy of protein structure identification is as follows:

$$PSIA = \frac{Number\ of\ features - Number\ of\ maximum\ Information\ gain}{Number\ of\ features} \times 100 \qquad (7)$$

Where PSIA represents the Protein Structure Identification Accuracy.

Table 3. Comparison of protein structure identification accuracy

Dataset	PSIA				
	PDB	proteinNet	PROSITE	SidechainNet	pfam
PSSP	83	84	86	88	90
HDX-NMR	88	89	90	92	93
TAPO	92	93	94	94	95
LR-ID3C	93	95	96	97	98
CP-CD	95	96	97	98	99

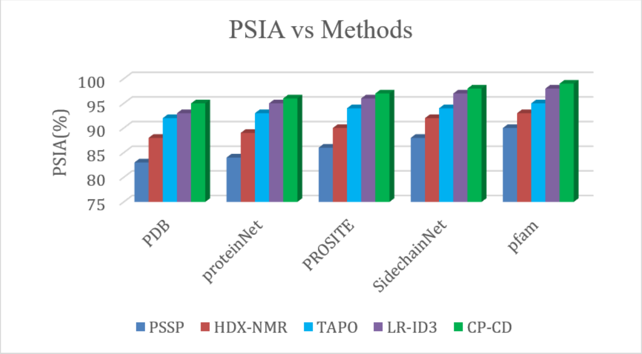


Figure 7. Comparison of PSIA for CP-CD method with existing techniques

Figure 7 shows the results of five protein datasets' protein structure identification accuracy. Table 3 shows that the proposed CP-CD technique produces higher protein structure identification accuracy results of 99% for 500 number of features (PDB), whereas existing methods such as LR-ID3C, TAPO, HDX-NMR, and PSSP produce protein structure identification accuracy results of 98 percent, 95 percent, 93 percent, and 90 percent, respectively.

c) False Positive Rate:

Table 4. Comparison of False positive rate

False Positive Rate					
Datasets	PDB	ProteinNet	PROSITE	SidechainNet	pfam
PSSP	24	22	20	19	15
HDX-NMR	23	20	19	17	13
TAPO	22	19	18	16	12
LR-ID3C	18	16	15	14	11
CP-CD	14	13	11	10	9

By dividing the number of features by the improperly detected features, one may calculate the false positive rate. It has a percentage (%) as its expression.

$$False\ Positive\ Rate = \frac{incorrectly\ identified\ feaures}{Number\ of\ features} \times 100 \qquad (8)$$

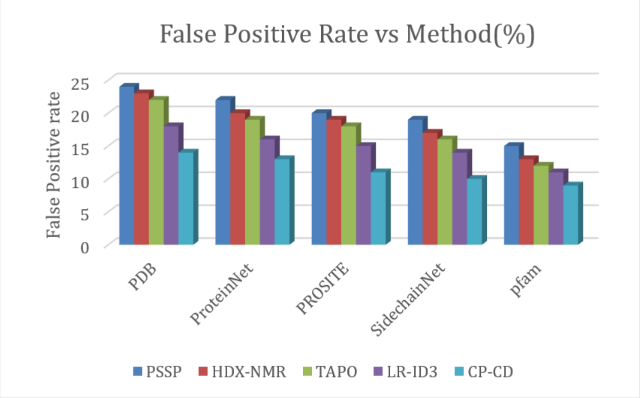


Figure 8. Comparison of False positive rate for CP-CD technique with other existing techniques

The false positive rate of the proposed CP-CD methodology and four existing techniques were compared in Figure 8. The suggested CP-CD methodology achieves a minimal false positive rate of 9% for 500 samples (PDB) as shown in the figure, but other techniques such as LR-ID3C, TAPO, HDX-NMR, and PSSP create false positive rates of 11%, 12 percent, 13 percent, and 15percent, respectively, as shown in Table 4.

d) Protein structure Identification Time

The time it takes to find a protein structure with the most information gain characteristics from a dataset is called protein structure identification time. The following is how time is calculated:

$$Protein\ structure\ identification\ Time = Number\ of\ features\ \times time \qquad (9)$$

Figure 9 shows the time it took five algorithms to identify protein structures on five protein datasets with varying numbers of characteristics.

Table 5. Comparison of protein structure identification time

Protein structure identification time					
Datasets	**PDB**	**ProteinNet**	**PROSITE**	**SidechainNet**	**pfam**
PSSP	25	27	29	31	33
HDX-NMR	23	24	26	28	29
TAPO	20	21	23	26	28
LR-ID3C	16	18	20	24	26
CP-CD	11	12	14	16	20

The proposed CP-CD technique produces a least protein structure identification time of 11 ms for 500 features (PDB) as shown in the figure, whereas other existing methods such as LR-ID3C, TAPO, HDX-NMR, and PSSP produce protein structure identification times of 16 ms, 20 ms, 23 ms, and 25 ms as shown in Table 5.

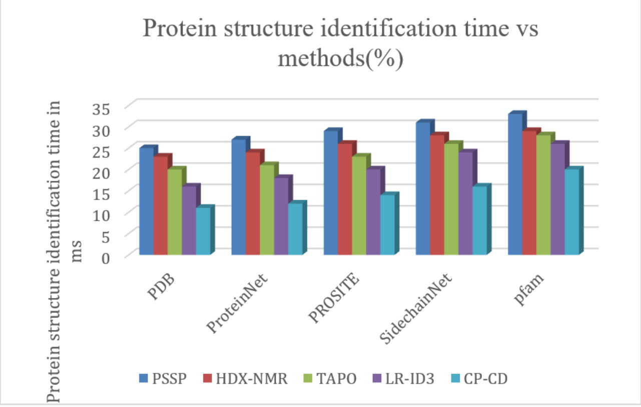


Figure 9. Comparison of Protein structure identification time for CP-CD technique with other existing methods

e) Precision

Divide the true positive rate by the total of the collection's true positive and false positive rates to get the precision. The exact mathematical formula is as follows:

$$precision(p) = \frac{True\ positive}{True\ positive+False\ positive} \qquad (10)$$

Table 6. Comparison of precision

Precision					
Datasets	**PDB**	**ProteinNet**	**PROSITE**	**SidechainNet**	**pfam**
PSSP	87	86	82	79	76
HDX-NMR	88	87	85	84	78
TAPO	89	87	84	83	80
LR-ID3C	92	90	89	85	87
CP-CD	94	91	90	89	90

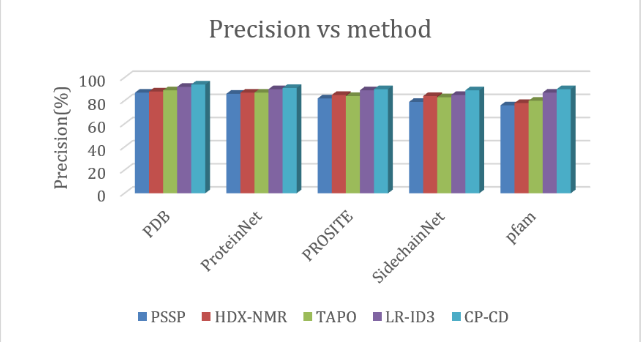


Figure 10. Comparison of Precision for CP-CD method with other existing methods

Figure 10 displays the accuracy findings of five different approaches. The proposed CP-CD method acquires higher precision results of 90% for 500 features (amino acid sequence from PDB), whereas other methods such as LR-ID3C, TAPO, HDX-NMR, and PSSP provide precision results of 87 percent, 80 percent, 78 percent, and 76 percent, respectively, as shown in Table 6.

f) Recall

The rate of number of relevant characteristics to the the absolute number of features that really correspond to the relevant features is how recall is calculated. The recall value is calculated in the following way:

$$Recall(R) = \frac{True\ positive}{True\ positive + False\ positive} \quad (11)$$

Table 7. Comparison of Recall

Recall					
Datasets	**PDB**	**ProteinNet**	**PROSITE**	**Sidechain Net**	**pfam**
PSSP	78	79	83	84	86
HDX-NMR	79	82	84	85	87
TAPO	82	83	84	86	89
LR-ID3C	84	85	86	89	90
CP-CD	85	86	87	90	92

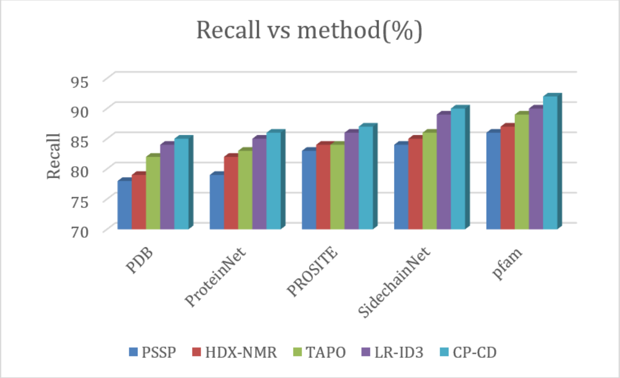


Figure 11. Comparison of Recall for CP-CD method with other existing methods

According to the Figure 11, the CP-CD method acquires larger recall results of 92 percent for 500 number of features of amino acid sequence from the PDB, whereas other methods such as LR-ID3C, TAPO, HDX-NMR, and PSSP produce recall results of 90 percent, 89 percent, 87 percent, and 86 percent values, which are mentioned in Table 7.

g) F-measure

The F-measure is a single positive class test measure. It's the weighted mean of a test's precision and recall. It is written down as follows:

$$F - measure = \frac{P \times R}{P + R} \quad (12)$$

Figure 12 shows that the proposed CP-CD method achieves higher F-measure results of 92 percent for 500 features (PDB), whereas other existing methods such as LR-ID3C, TAPO, HDX-NMR, and PSSP produce F-measure results of 90 percent, 88 percent, 87 percent, and 86 percent, respectively, as shown in Table 8.

Table 8. comparison of F-measure

F-measure					
Datasets	**PDB**	**Protein Net**	**PROSITE**	**SidechainNet**	**pfam**
PSSP	76	78	79	83	86
HDX-NMR	78	82	84	85	87
TAPO	79	83	85	87	88
LR-ID3C	82	84	86	89	90
CP-CD	85	87	88	90	92

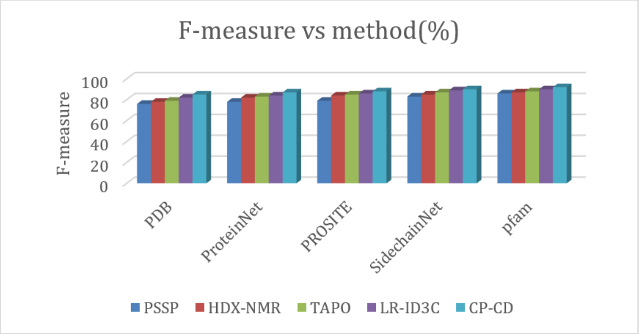


Figure 12. Comparison of F-measure for CP-CD method with other existing methods

5. CONCLUSION

In biotechnology, determining the function of proteins is a critical step. Because of the time necessary to execute the categorization tasks, the excessive characteristics render computer systems unproductive. As a result, determining protein structure based on amino acid sequence requires careful attention. Therefore, a method called CSA-PSO based CD4.5 Classification technique is used. Here, the samples of the patients are collected and tested through an IOT enabled microsope and the details will be stored in the big data cloud.Then it have two steps: Initially, feature selection will takes place through a hybrid crow search algorithm and particle swarm optimization algorithm(CSA-PSO). It aids in increasing the chances of specific outcomes. This aids in improving the true positive rate in a short amount of time. Later, classification is done by using CD4.5 classifier, which is an extension of ID3 classifier. For making a choice to identify the structure, characteristics with the least entropy are picked. This improves the accuracy of protein structure recognition and lowers the percentage of false positives. Different protein datasets are used to evaluate the CP-CD approach in the experiment. When compared to the other protein dataset samples, PDB datasets provide better performance results. When comparing to other techniques, the suggested CP-CD methodology considerably enhances true positive rate, protein structure identification accuracy, precision, recall, F-measure with minimal protein structure identification time, and false positive rate. Further studies will be taken to use the offered approaches to deal with issues in protein structure identification and a high-dimensional perspective of structure in the future.

CONFLICTS OF INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

FUNDING STATEMENT

Not applicable.

ACKNOWLEDGEMENTS

The author would like to express his heartfelt gratitude to the supervisor for his guidance and unwavering support during this research for his guidance and support.

REFERENCES

[1] N. Eswar, D. Eramian, B. Webb, M.Y. Shen, and A. Sali, "Protein structure modeling with Modeller", *In Structural proteomics*, pp. 145-159, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[2] M.M. Gromiha, "Protein bioinformatics: from sequence to function", academic press. 2010. [CrossRef] [Google Scholar] [Publisher Link]

[3] K. Kira, and L.A. Rendell, "A practical approach to feature selection", *In Machine learning proceedings* 1992, pp. 249-256, 1992. [CrossRef] [Google Scholar] [Publisher Link]

[4] A.R.H. Hussein, "Internet of things (IOT): Research challenges and future applications", *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 6, pp.77-82, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[5] J. Masood, M. Shahzad, Z.A. Khan, V. Akre, A. Rajan, S. Ahmed, and F. Masood, "Effective Classification Algorithms and Feature Selection for Bio-Medical Data using IoT," *In 2020 Seventh International Conference on Information Technology Trends (ITT)*, IEEE pp. 42-47, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[6] A.M. Sequeira, D. Lousa, and M. Rocha, "ProPythia: A Python package for protein classification based on machine and deep learning", *Neurocomputing*. 2021. [CrossRef] [Google Scholar] [Publisher Link]

[7] B. Kalaiselvi, and M. Thangamani, "An efficient Pearson correlation based improved random forest classification for protein structure prediction techniques", *Measurement*, Vol. 162, pp.107885, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[8] Y. Ge, S. Zhao, and X. Zhao, "A step-by-step classification algorithm of protein secondary structures based on double-layer SVM model", *Genomics*, vol. 112, no. 2, pp. 1941-1946, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[9] J.J. Shu, and K.Y. Yong, "Fourier-based classification of protein secondary structures", *Biochemical and biophysical research communications*, vol. 485, no. 4, pp.731-735, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[10] S.M. Najibi, M. Maadooliat, L. Zhou, J.Z. Huang, and X. Gao, "Protein structure classification and loop modeling using multiple Ramachandran distributions", *Computational and structural biotechnology journal*, vol. 15, pp.243-254, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[11] J. Ahmad, and M. Hayat, "MFSC: Multi-voting-based feature selection for classification of Golgi proteins by adopting the general form of Chou's PseAAC components", *Journal of Theoretical Biology*, vol. 463, pp. 99-109, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[12] A. Ahmad, S. Akbar, M. Hayat, F. Ali, and M. Sohail, "Identification of antioxidant proteins using a discriminative intelligent model of k-space amino acid pairs-based descriptors incorporating with ensemble feature selection", *Biocybernetics and Biomedical Engineering*, 2020. [CrossRef] [Google Scholar] [Publisher Link]

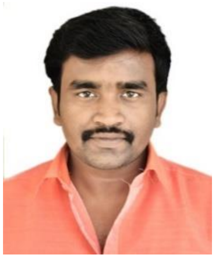
[13] G. Mirceva, A. Naumoski, and A. Kulakov, "Classification of Protein Structures by Making Fuzzy-Rough Feature Selection", *In 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pp. 1-42020. [CrossRef] [Google Scholar] [Publisher Link]

[14] G. Mirceva, I. Ivanoska, A. Naumoski, and A. Kulakov, "Feature Selection for Improved Classification of Protein Structures", *In 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1013-1018, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[15] K.K. Ghosh, S. Ghosh, S. Sen, R. Sarkar, and U. Maulik, "A two-stage approach towards protein secondary structure classification", *Medical & Biological Engineering & Computing*, vol. 58, pp. 1723-1737, 2020. [CrossRef] [Google Scholar] [Publisher Link]

AUTHORS

T. Maris Murugan Associate Professor and Head, Department of Electronics and Instrumentation Engineering, at Erode Sengunthar Engineering College, Erode, Tamilnadu, India. He was awarded B.E. degree in Electronics and Instrumentation Engineering from Government College of Technology, Coimbatore and M.E. degree in Embedded Systems from Karpagam University Coimbatore, Tamilnadu, India. He was awarded Ph.D. degree from Anna University, Chennai, Tamilnadu. He has 14 years in Industry and as well as 15 years in academic experience with expertise in the field of Process Control, Industrial Automation, Machine Learning and Internet of Things.



A. Jeyam received his Master Degree (M.E) in Computer Science and Engineering GCT-Coimbatore during 2012-2014. He has very good knowledge in Computer related subjects like Java, Python, Database systems, Foundations of Computer Systems, Website Designing, Software Engineering, Database Management Systems, Object Oriented Programming, Image Processing and Data Mining in university and college level, good knowledge in Computer software, Hardware and Networking. He has done many projects in Image Processing. He has worked as a Lecturer and Network Administrator for more than 05 years in Research Point India.

Arrived: 29.11.2023

Accepted: 28.12.2023