BIOCAD: Bio-Inspired Optimization for Classification and Anomaly Detection in Digital Healthcare Systems

Nur Imtiazul Haque*†, Alvi Ataur Khalil*†, Mohammad Ashiqur Rahman†,
M. Hadi Amini‡, Sheikh Iqbal Ahamed§
†Analytics for Cyber Defense (ACyD) Lab, Florida International University, FL, USA
‡Knight Foundation School of Computing and Information Sciences, Florida International University, FL, USA
§Department of Computer Science, Marquette University, WI, USA
†{nhaqu004, akhal042, marahman}@fiu.edu, ‡moamini@fiu.edu §sheikh.ahamed@marquette.edu

***Abstract*—The modern smart digital healthcare system (SDHS) is leaning towards automation of patient disease monitoring and treatment with the advent of wireless body sensor networks (WBSN) and the internet of medical things (IoMT). However, the open communication network for sensitive medical data transfer is giving rise to vulnerabilities and security concerns. To prevent adversarial manipulation of sensor measurements, SDHS IoMT controllers leverage anomaly detection systems on top of the disease classification systems. Machine learning (ML) is one of the most effective techniques for providing experience-based automated decision-making models. These models generalize well to produce the expected output for the unseen inputs from the learned patterns. Therefore, ML-based models are currently being adopted to automate the anomaly detection and disease classification tasks of SDHS. In this work, we consider a SDHS that uses supervised ML models for patient status/disease classification and unsupervised ML models for anomaly detection. However, the performance of the ML models largely depends on hyper-parameter tuning. Finding the optimal hyper-parameter is a challenging task, and it becomes more difficult and time-consuming in high-dimensional feature space. In this work, we propose BIOCAD, a comprehensive bio-inspired optimization framework for SDHS data classification and anomaly detection. The framework leverages a novel fitness function for unsupervised anomaly detection ML models. We experiment with state-of-the-art datasets - the Pima Indians diabetes dataset, the Parkinson dataset, and the University of Queensland vital signs (UQVS) dataset for validating our proposed strategy.**

***Index Terms*—Healthcare Security, machine learning, anomaly detection systems, hyperparameter Optimization**

I. Introduction

The world is currently fighting with COVID-19, a lightless adversary and every day, thousands of people are getting affected by it throughout the world. The healthcare consultants, nurses, and volunteers are fighting heart and soul to ensure treatment for all patients. However, due to the contagious nature of the virus, the frontliners are also getting affected, which increases the number of patients even more. The need for global acceptance of a smart digital healthcare system (SDHS)-enabling automated medication, treatment, and pill dispensers is obvious. Besides, the healthcare sector is experiencing high costs due to patient monitoring, consultation, hospitalization, and treatment, which is proven from almost $4 trillion patient treatment expenditure in the US as reported in 2019 [1]. To reduce this cost and ensure pervasive healthcare, the modern SDHS uses the Internet of Medical Things (IoMT) network to collect sensor measurements from wireless body sensor network (WBSN) and process them in a controller for generating control signals, which eventually actuates the implantable medical devices (IMDs) for automated medical delivery [2]. Identifying the right disease is mandatory for delivering proper medication. Hence, machine learning (ML)-based classification model (CM) is mostly used in the controller end of the contemporary SDHS for their high prediction accuracy.

The SDHS sensors cannot be blindly trusted due to the increasing vulnerability and attack surface of the WBSN. Most popular cyberattacks in the healthcare systems includes hardware Trojan [3], malware (e.g., Medjack [4]), Sybil attacks using either hijacked IoMT [5] or single malicious node [6], DoS attacks [7], and man-in-the-middle (MITM) attacks [8]. Recent statistics show that cyberattacks in SDHS are surging and exploiting the security and privacy of the patients. Healthcare organizations are experiencing a 45-percent increase in cyberattacks with 626 weekly attacks since November 2020 [9], [10]. In June 2020, a hospital in Colorado was affected by a ransomware attack which resulted in five years worth of patient data inaccessible [10]. The University of Vermont Medical Center (UVMC) is losing $1.5 million per day in revenue and extra expenses due to recent cyberattacks and estimates to $63 million more before resolving the issues [11]. In addition, a security breach in the Blackbaud cloud service provider made more than 46 hospitals and health systems expose 1 million patients information. It is imperative to identify sensor data alteration to save patients' life. Hence, detecting cyberattacks is mandatory in a safety-critical system like SDHS, which creates the need for an anomaly detection model (ADM) with zero-day attack detection capability.

Current research focuses on getting the best out of the ML

*Haque and Khalil are the co-first authors of this paper.

algorithms by finding the optimal hyperparameters using various optimization techniques [12]. Optimization of ML refers to adjusting the underlying hyperparameters to minimize a cost function using an appropriate optimization model (OM). In this work, we propose **B**io-**I**nspired **O**ptimization for **C**lassification and **A**nomaly **D**etection (BIOCAD) framework that optimizes CM and ADM of the SDHS using OMs. Our proposed framework utilizes ML-based ADMs that detects alteration in SDHS data before providing automated medication and alarms the system administrators for taking necessary measures.

Methods for solving optimization problems are an active research topic. New optimization algorithms can be either deterministic or stochastic. The deterministic algorithms produce identical output as the input. On the other hand, the stochastic algorithms incorporate pseudorandomness for learning uncertainties [13]. As deterministic algorithms require massive computational efforts to solve optimization problems, they fail with increased problem size. Hence, the motivation behind choosing bio-inspired stochastic optimization algorithms is the computational efficiency of these models, unlike deterministic approaches. Although state-of-the-art problem-solving strategies include both the Exact approach (i.e., logical programming) and the Heuristic approach, the latter performs better in solving hard and complex optimization problems [14]. The bio-inspired algorithms (BA) are unique heuristic approaches that imitate nature's strategy as a process of constrained optimization.

Our proposed framework leverages some of the prominent algorithms from the Swarm-based algorithms for hyperparameter optimization, as they complete the tasks even if some of the agents fail. Also, there is no central control, and the solutions are more emergent rather than pre-defined. In this work, we consider grey wolf optimization (GWO), whale optimization (WO), and firefly optimization (FO) algorithms for optimizing our support vector machine (SVM), neural network (NN)- based CM, and one-class SVM (OCSVM), Density-based spatial clustering of applications with noise (DBSCAN)-based ADM models to assess the performance of our framework.

In summary, our contributions are four-fold:

- Investigating various ML algorithms, we design a real-time SDHS deploying ML-based CMs and ADMs.
- We evaluated the proposed BIOCAD framework based on it's classification, anomaly detection capability and scalability.
- We propose a novel fitness function for hyper-parameter optimization of ADMs to detect zero-day attacks.
- We verify our proposed framework using three state-of-the-art datasets.

Organization: The rest of the paper is organized as follows: the recent related works are discussed in Section II. We discuss adequate background information in Section III. We introduce our proposed BIOCAD framework in Section IV. In Section V, we discuss the technical details of the frameworks and the complete analysis of our algorithms. The empirical analysis and findings are formulated in Section VI. Finally, we conclude the paper in Section VII.

II. Related Works

Safety critical systems like SDHS are getting a lot of research focus. The researches include attack synthesis, threat analysis, and proposing IDSs [15]–[18]. Contemporary CMs and ADMs leverage several mechanisms to optimize the hyperparameters of the underlying ML models. The hyperparameter optimization techniques reduce the overfitting problem and enhance the accuracy of the models. Dwivedi et al. proposed an intrusion detection system by utilizing the grasshopper optimization algorithm (GOA), where an ensemble feature selection method is used for ranking the top features before optimization [19]. They obtained high detection and accuracy rate as well as a low false alarm rate in NSL-KDD and KDD Cup 99 data. Falaghi et al. proposed an ant colony optimization (ACO)-based classification model for solving the fuzzy multi-objective problem, which can produce better subsets and achieve higher classification accuracy [20].

Several researchers attempted to ameliorate the performance of SVM for medical data classification by proposing novel BAs. Shen et al. proposed a novel fruit fly algorithm (FFA) for both parameter and classification optimization of SVM and obtained 96%-97% accuracy for various medical data classification [21]. On average, their proposed approach has taken 170 seconds of CPU time to complete the optimization task. Their evaluation result shows that the proposed FFA algorithm works significantly better than PSO, GA, BFO, and grid search technique, concerning execution time and performance measure. Ye et al. proposed a novel GA, combining PSO and FOA [22]. Their algorithm shows 98%-99% accuracy for medical data diagnosis. Wang et al. proposed a modified WO algorithm, amalgamating muti-swarm and chaotic mechanisms, and found accuracy in between 98%-99% for several medical datasets [23]. Contrasting with GA, BFO, WO, and PSO, they have shown that the proposed algorithm provides better performance and scalable execution time.

Al Shorman et al. proposed a novel mechanism for detecting IoT botnet attacks using OCSVM and utilized GWO to optimize the underlying hyperparameters in [24]. The experimental results confirm that the proposed GWO-OCSVM outperforms regular OCSVM, Local Outlier Factor (LOF), and Isolation Forest (IF) algorithms concerning true positive rate, false-positive rate, and geometric mean. Ch et al. optimized the parameters of SVM for determining and forecasting the incidence of malaria in [25] by coupling it with FO. They compared the performance of the proposed FO-SVM with regular SVM, Auto-Regressive Moving Average (ARMA), and Artificial Neural Networks (ANN) method, and the results show that FO-SVM can forecast the incidences more accurately than the other methods.

All the anomaly detection works mentioned above consider known attack data, often along with benign data, to identify anomalies. Despite our ADM is solely trained on benign data,

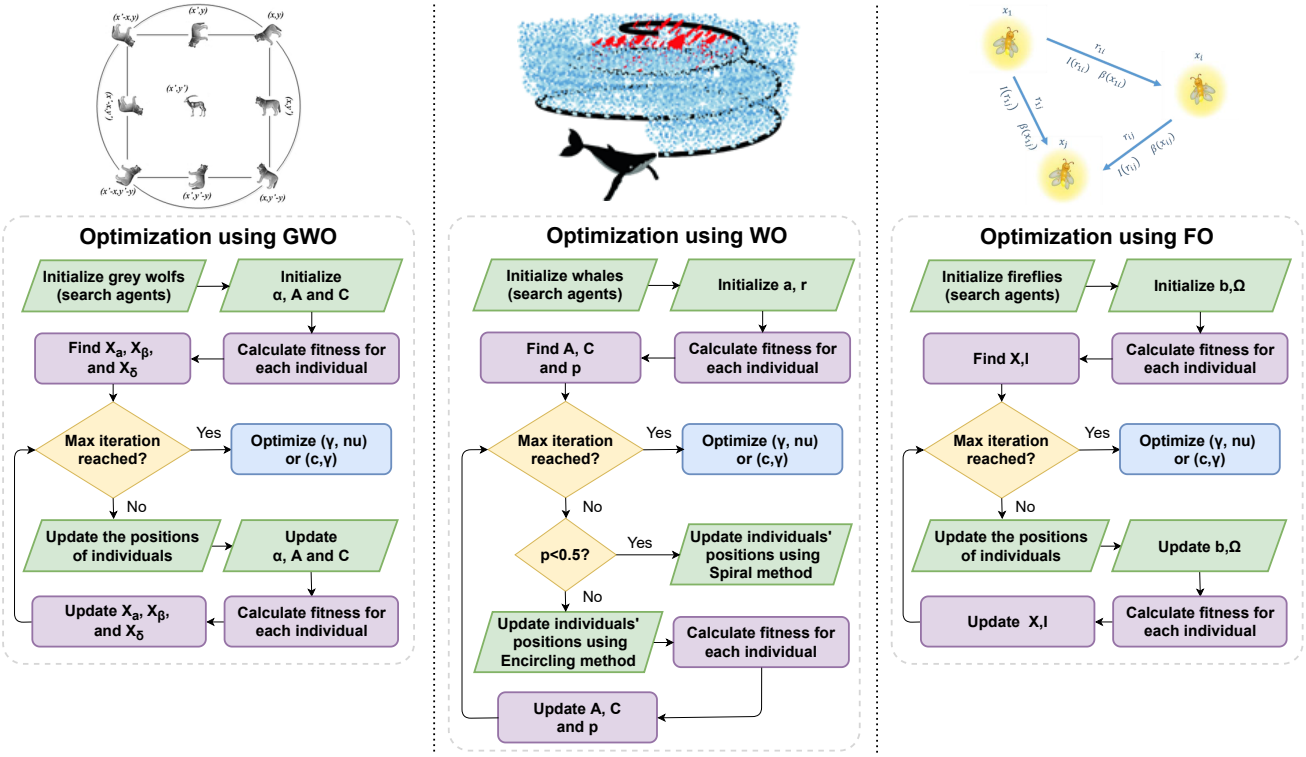


Fig. 1. Schematic diagram of the optimization algorithms of the BIOCAD framework.

it performs significantly well on anomaly/zero-day attack detection. In our approach, we leverage several BAs to optimize the hyperparameters of the underlying ML models of our ADMs and CMs.

III. Preliminaries

Several ML models and OMs are leveraged in this work. For anomaly detection, our framework uses a novelty detection ML model- OCSVM and a clustering-based ML model- DBSCAN, while for disease/patient status prediction, the framework leverages two supervised ML models- SVM and NN. Additionally, three state-of-the-art bio-inspired optimization algorithms - FO, WO, and GWO are utilized in this work for optimizing the hyper-parameters of the considered ML models. In this section, we provide some insights into the considered models to facilitate the readers' comprehension.

A. One-Class Support Vector Machine

The OCSVM solves a one-class classification problem by splitting the target instances from outlier instances [26]. Let, x_i, i = 1, ... , n are the training instances in the input space. OCSVM works based on the principle of differentiating the outlier samples from the target ones. Suppose, d_i, i = 1, .., n are benign training examples in $\mathcal{D}$ input space, where, $\psi(d_i)$ is the transfer function for non-linearly mapping an sample d_i from $\mathcal{D}$ into $\mathbb{F}$ feature space. The main objective of OCSVM is to build a hyperplane in the feature space, $\mathbb{F}$ to maximize a margin from the origin. Lets say, $\mathbb{H} :< w, \psi(d) > -\mathcal{T} = 0$ resembles the hyperplane, where, w is denotes the normal to the hyperplane, $\mathbb{H}$ and $\mathcal{T}$ is a threshold. The threshold $\mathcal{T}$ can be determined by solving the following optimization function.

$$\min_{w\in\mathbb{F},\xi\in\mathbb{R}^n,\mathcal{T}\in\mathbb{R}} \frac{1}{2}||w||^2 + \frac{1}{\nu n}\sum_i \xi_i - \mathcal{T} \tag{1}$$

subject to, $< w, \psi(d_i) > \geq \mathcal{T} - \xi_i$.

Here, ξ are called the slack variables, which are used to impose an equality constraint from an inequality one. The ν parameter indicates the upper and lower bounds on the percentage of outliers in all samples used as support vectors. Based on the Lagrangian dual problem, the OCSVM's dual problem can be obtained. It can be formulated as follows:

$$\max_a -\frac{1}{2}\sum_{i,j=1}^{n} \mu_i\mu_j k(d_i, d_j) \tag{2}$$

such that

$$0 \leq \mu_i \leq \frac{1}{\nu n} \tag{3}$$

and

$$\sum_{i=1}^{n} \mu_i = 1 \tag{4}$$

Here, $k(d_i, d_j)$ corresponds to the kernel function, which outcomes same value as $< \psi(d_i, d_j) >$. After obtaining the optimal solution, μ, putting $\mathcal{T} =< w, \psi(d_i) >$, the threshold can be determined, where, $w = \sum_{i=1}^{n} \mu_i\psi(d_i)$ and d_i denote a samples whose $\mu_i \in (0, \frac{1}{\nu n})$. The samples, with $\mu_i > 0$ are referred support vectors. The OCSVM decision function kf(d) can be obtained by using appropriate kernel function, as shown below:

$$kf(d) =< w, \psi(d_i) > - = \sum_{i=1}^{n} \alpha_i k(d_i, d) - \mathcal{T} \tag{5}$$

For benign (positive) samples, the decision function comes up with a positive or zero value, while anomalous (negative) samples show a negative outcome.

B. DBSCAN

The DBSCAN algorithm demonstrates optimistic performance for identifying abnormality in the data based on the deviation from learned pattern [27]. The DBSCAN algorithm uses a pair of hyperparameters- the minimum number of points ($minPts$) and epsilon(ϵ). At first, the algorithm attempts to find the points in the epsilon neighborhood of every point. The points having more than $minPts$ neighbors in the ϵ radius are denoted as core points. Then the components of core points on the neighbor graph are identified (all non-core points are ignored in this phase). Finally, the DBSCAN algorithm assigns each non-core point to a nearby cluster if the cluster is an ϵ neighbor; otherwise, it assigns it to noise. In this work, we consider the noise points as anomalous data.

C. Support Vector Machine

SVM is a supervised learning algorithm, which is used for separating different classes by drawing hyper-planes like OCSVM. Unlike OCSVM, this algorithm creates the hyperplane to differentiate the support vectors of two different classes. For solving a multi-class classification problem, SVM uses a one-vs-all approach [28]. A specialized technique referred to as kernel trick is applied to obtain the large margin hyperplanes for constructing nonlinear classifiers. The technique is formally similar to the simpler version, with a minor modification of the dot products by a nonlinear kernel function, enabling the modified SVM to adjust the maximum-margin hyperplane in a new feature space. The SVM model's optimal decision boundary can be obtained by tuning two hyperparameters- γ and ν.

D. Neural Network

NN refers to the ML model that learns nonlinear patterns from myriad feature relationships set. It contains a network of nodes, where the nodes are arranged according to particular ways depending on the application. Patterns include layers, starting from the input layer to the output layer and the variable number of hidden layers in between. The number of hidden layers and the number of nodes in each hidden layer play a crucial role in the neural network's performance. For each node of the NN model, the sum of products of weight and output of the previous nodes added with the bias work as the input. To add non-linearity in the model, the output of each node is passed through an activation function. For drawing nonlinear boundaries in the feature space to non-linearly map between the input features and target, activation function plays a vital role. Learning rate is the hyperparameter that controls the updating magnitude of the model parameters in response to the estimated error. We leveraged RMSprop [29] for adaptively choosing and tuning the earning rate. For generalizing the model and reducing overfitting with the training samples, the regularization process is also applied. For achieving the optimal nonlinear boundaries from the NN model, the number of nodes and the number of hidden layers have to be tuned.

E. Grey Wolf Optimization Algorithm

Grey wolf optimization (GWO) is one of the recent prominent meta-heuristic optimization algorithms presented by Mirjalili et al. in [30]. This optimization algorithm is modeled after the two most important survival aspects of the grey wolves, the hunting mechanism and the leadership hierarchy. For simulating the leadership hierarchy, four varieties of grey wolves, namely the α, β, δ, and ω wolves are used. The hunting of grey wolves pack has three keys stages- looking for prey, encircling prey, and attacking prey. This is also simulated in the GWO algorithm. The mathematical model of the GWO algorithm is described in the subsequent paragraphs.

For mathematically modeling the social hierarchy of the wolves pack, multiple candidate solutions are considered. The best-observed solution is considered as the α. Consequently, the second and third observed optimal solutions are called β and δ. The ω wolf represents the rest of the candidate solutions. In the GWO algorithm, α, β, and δ wolves (solutions) guide the hunting scheme (optimization). The ω follow the three most optimal wolves (solutions).

The grey wolves encircle the target prey at time of hunting. Following equations are proposed for mathematically modeling the encircling behavior:

$$\vec{\mathcal{D}}^{grw} = |\vec{\mathcal{L}}^{grw}.\vec{\mathcal{X}_{Pr}}(t) - \vec{\mathcal{X}}(t)| \tag{6}$$

$$\vec{\mathcal{X}}^{grw}(t+1) = \vec{\mathcal{X}_{Pr}}(t) - \vec{\mathcal{K}}^{grw}.\vec{\mathcal{D}}^{grw} \tag{7}$$

Here, t denotes the current iteration, $\vec{\mathcal{K}}^{grw}$ as well as $\vec{\mathcal{L}}^{grw}$ signifies the coefficient vectors, $\vec{\mathcal{X}_{Pr}}$ corresponds to the prey's position vector, $\vec{\mathcal{X}}$ represents the current position vector and finally, using $\vec{\mathcal{X}}^{grw}$, GWO determines new position vector of a grey wolf. The co-efficient vectors $\vec{\mathcal{K}}^{grw}$ and $\vec{\mathcal{L}}^{grw}$ follows the following equation:

$$\vec{\mathcal{K}}^{grw} = 2\vec{k}.\vec{r_1} - \vec{k} \tag{8}$$

$$\vec{\mathcal{L}}^{grw} = 2.\vec{r_2} \tag{9}$$

The $\vec{k}$ vectors' component are linearly altered from 2 to 0 over iterations and $\vec{r_1}$, $\vec{r_2}$ are random vectors ($0 \leq r_1, r_2 \leq 1$). The similar idea can be utilized to search upto n dimension space, and the grey wolves will follow the best solution agent and will move around it in the hyper-spheres.

Grey wolves possess the ability to track and encircle prey. In an arbitrary search space, though, there is no information about where the optimum (prey) can be found. It is believed that α (the best agent solution), β, and δ have greater knowledge of the possible position of prey. It is actually the mathematical replication of the stalking grey wolves' behavior. Therefore, the current best three solutions are stored and the other candidates update their positions based on the position of the best search agents. The proposed equations in this regard are:

$$\vec{\mathcal{D}_\alpha}^{grw} = |\vec{\mathcal{L}_1}^{grw}.\vec{\mathcal{X}_\alpha} - \vec{\mathcal{X}}|, \vec{\mathcal{D}_\beta}^{grw} = |\vec{\mathcal{L}_2}^{grw}.\vec{\mathcal{X}_\beta} - \vec{\mathcal{X}}|, \\ \vec{\mathcal{D}_\delta}^{grw} = |\vec{\mathcal{L}_3}^{grw}.\vec{\mathcal{X}_\delta} - \vec{\mathcal{X}}| \tag{10}$$

$$\vec{\mathcal{X}_1}^{'} = \vec{\mathcal{X}_\alpha} - \vec{\mathcal{K}_1}^{grw}.(\vec{\mathcal{D}_\alpha}^{grw}), \vec{\mathcal{X}_2}^{'} = \vec{\mathcal{X}_\beta} - \vec{\mathcal{K}_2}^{grw}.(\vec{\mathcal{D}_\beta}^{grw}), \vec{\mathcal{X}_3}^{'} = \vec{\mathcal{X}_\delta} - \vec{\mathcal{K}_3}^{grw}.(\vec{\mathcal{D}_\delta}^{grw}) \tag{11}$$

$$\vec{\mathcal{X}}^{grw}(t+1) = \frac{\vec{\mathcal{X}_1}^{'} + \vec{\mathcal{X}_2}^{'} + \vec{\mathcal{X}_3}^{'}}{3} \tag{12}$$

The hunting step is terminated by committing the attack as soon as the prey becomes stationary. The value of $\vec{k}$ is decreased for mathematically modeling this approach. This also decreases the fluctuation range of $\vec{\mathcal{K}}^{grw}$ by k. So, $\vec{\mathcal{K}}^{grw}$ is a randomly chosen value in the interval [-2k, 2k]. Also, with the increase of iterations, the value of parameter a decreases from 2 to 0. When $\vec{\mathcal{K}}^{grw}$'s random values are in the interval of [-1,1], the next position of the agent can be anywhere between its current position and the position of the optimal solution (prey). The wolves are obliged to attack the prey when $|\mathcal{K}^{grw}| < 1$.

Grey wolves search mainly using the α, β, and δ positions. They isolate to hunt for prey and then unite to attack the prey. A random value out of the [-1,1] range is used with $\vec{\mathcal{K}}^{grw}$ for ensuring model divergence, which enables search agents to diverge from the prey. +1 or less than -1 is utilized to force the search agent to diverge from the prey. This encourages exploration and enables the GWO algorithm to search around the global space. $\vec{\mathcal{L}}^{grw}$ is another GWO aspect that promotes exploration. The $\vec{\mathcal{L}}^{grw}$ vector contains an arbitrary value ranging between 0 to 2. Random wights are assigned to the prey by this parameter to emphasize ($\mathcal{L}^{grw} > 1$) or deemphasize ($\mathcal{L}^{grw} < 1$) the prey's influence in deciding the distance stochastically. This ensures that the GWO will have random behavior during the optimization, preferring exploration and avoiding the local optima problem.

F. Whale Optimization Algorithm

Mirjalili et al. proposed the WO algorithm based on the unique foraging behavior of the humpback whales [31]. The humpback whales target the small fishes swimming near the water surface, and this hunting behavior is termed bubble-net feeding. They create '9' or circle-shaped bubbles for hunting, which are named as 'double loops' and 'upwards spiral respectively [32]. The mathematical representation of the behavior serves as a meta-heuristic optimization algorithm, which can be leveraged to solve various intricate problems.

The WO algorithm is a three-stage process in which the prey encircling phase identifies the prey's location, bubble-net attacking is the exploitation phase. In contrast, prey searching is the exploration phase for finding the optimal prey location. Using a similar strategy, the global optima of an optimization problem can be searched. The mathematical modeling of the three phases of the WO algorithm is discussed as follows.

1) Prey encircling: In the prey encircling phase, WO finds out the location of the prey assuming close proximity of the target. Several humpback whales participate in this process and the path of the best agent is followed by the rest. Equation 13 illustrates the mathematical form of the aforementioned behaviour.

$$\vec{\mathcal{D}}^{wh} = |\vec{\mathcal{C}}^{wh}.\vec{\mathcal{X}}^{wh}_{best}(t)| - \vec{\mathcal{X}}^{wh}(t) \tag{13}$$

$$\vec{\mathcal{X}}^{wh}(t+1) = \vec{\mathcal{X}}^{wh}(t+1) - \vec{\mathcal{A}}.\vec{\mathcal{D}} \tag{14}$$

Here, $\vec{\mathcal{X}}^{wh}_{best}(t)$ represents the position of the best agent in the current iteration, $\vec{\mathcal{X}}^{wh}(t)$ denotes a position vector, and the co-efficient vectors $\vec{\mathcal{A}}$ and $\vec{\mathcal{C}}$ can be determined by the following equations:

$$\vec{\mathcal{A}}^{wh} = 2\vec{\mathcal{A}}^{wh}.\vec{r}^{wh} - \vec{\mathcal{A}}^{wh} \tag{15}$$

$$\vec{\mathcal{C}}^{wh} = 2.\vec{r}^{wh} \tag{16}$$

Where, $\vec{\mathcal{A}}^{wh}$ is linearly altered from 2 to 0 and $\vec{r}^{wh}$ corresponds to a random vector ($0 \leq \vec{r}^{wh} \leq 1$). It can be proven that regulating $\vec{\mathcal{A}}^{wh}$ and $\vec{\mathcal{C}}^{wh}$, overall search space near agents can be explored.

2) Bubble-net attacking: Bubble-net attacking is the exploitation phase, which follows two different mechanisms.

Shrinking encircling In the shrinking encircling mechanism, $\vec{\mathcal{A}}$ is randomly adjusted in the interval [-a, a]. The value of a is lowered linearly from 2 to 0 over time. This process allows a linear movement throughout iterations.

Spiral updating position The spiral updating equation imitate a helix-shape progression of the agents, which can be represented as:

$$\vec{\mathcal{X}}^{wh}(t+1) = \vec{\mathcal{D}'}^{wl}.e^{bl}.cos(2\pi l) + \vec{\mathcal{X}}^{wh}_{best}(t) \tag{17}$$

Here, $\vec{\mathcal{D}'}^{wl} = |\vec{\mathcal{X}}^{wh}_{best}(t) - \vec{\mathcal{X}}^{wh}(t)|$, which denoted the distance from prey for i-th whale, b is a logarithmic spiral shape constant, and $-1 \leq l \leq 1$ is a random value.

Both of the aforementioned mechanisms are simultaneously followed based on a probability value, p.

3) Prey searching: Prey searching is the exploration phase, which can be attained by varying $\vec{\mathcal{A}}^{wh}$. The value of $\vec{\mathcal{A}}$ is out of [-1,1] interval, which ensures exploring remote places from the reference agent. The mathematical form of exploration can be expressed as follows.

$$\vec{\mathcal{D}}^{wh} = |\vec{\mathcal{C}}^{wh}.\vec{\mathcal{X}}^{wh}_{rand} * (t) - \vec{\mathcal{X}}^{wh}(t)| \tag{18}$$

$$\vec{\mathcal{X}}^{wh}(t+1) = \vec{\mathcal{X}}^{wh}_{rand}(t+1) - \vec{\mathcal{A}}^{wh}.\vec{\mathcal{D}}^{wh} \tag{19}$$

Here, $\vec{\mathcal{X}}^{wh}_{rand}$ is a random vector selected from the current population.

G. Firefly Optimization Algorithm

FO is another bio-inspired algorithm for solving complex optimization problems, developed by Xin-She Yang [33]. The algorithm is established based on the behavior of the fireflies, which approaches towards the direction depending on the luminosity of other fireflies. The FO algorithm utilizes three basic principles.

- The attraction of fireflies towards other brighter ones does not dependent on sex as all fireflies are unisex.
- The encoded objective function determines the brightness of the fireflies.

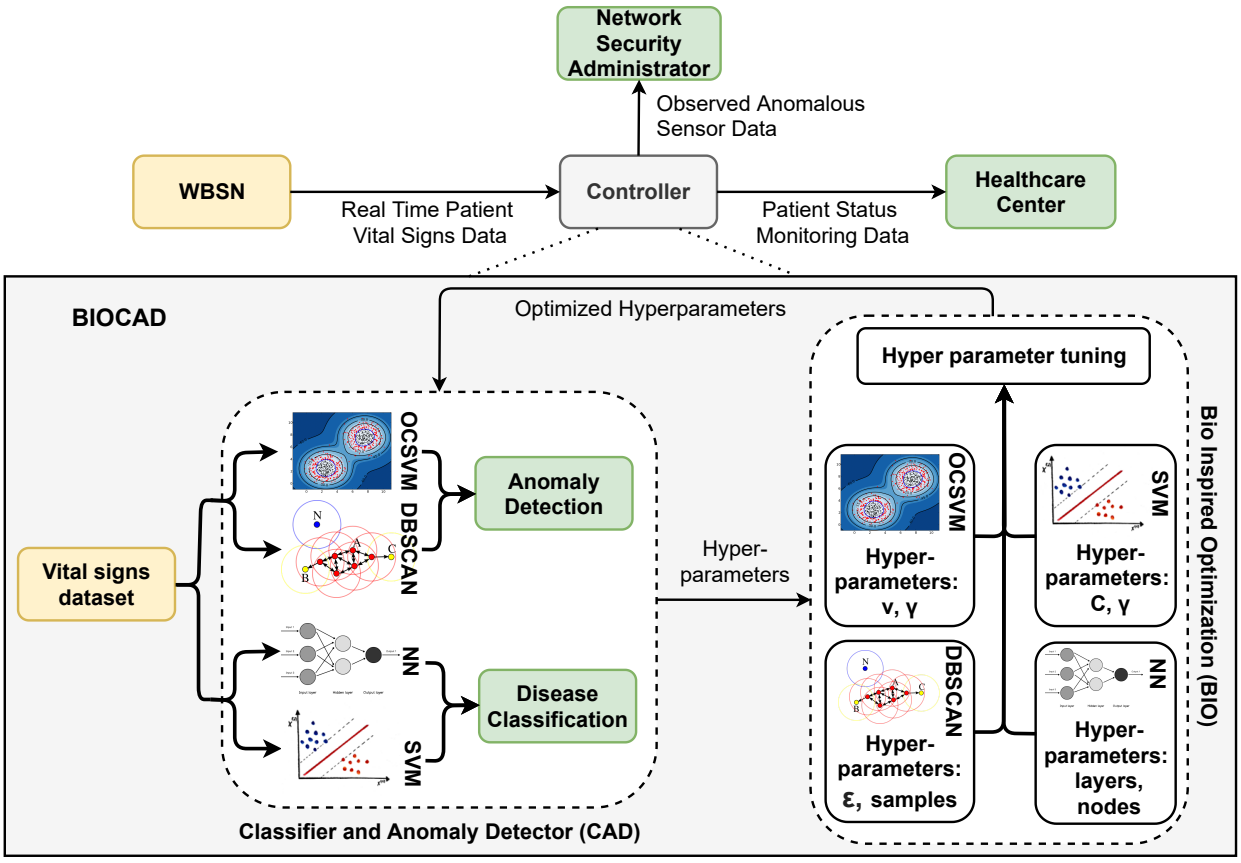


Fig. 2. Proposed BIOCAD framework.

- The brightness and attractiveness decrease with the increment in the distance, and the attractiveness directly corresponds to brightness.

In summary, a firefly moves towards the brighter one, and if there is no brighter one, it moves randomly. The mathematical representation position update of the firefly maintains the following equation:

$$\vec{\mathcal{X}}_i^{ff}(t+1) = \vec{\mathcal{X}}_i^{ff}(t) + \gamma^{ff} e^{-\beta r_{ij}^2}(\vec{\mathcal{X}}_j^{ff}(t) - \vec{\mathcal{X}}_i^{ff}(t)) + \alpha^{ff}(t)\epsilon_i^{ff}(t) \tag{20}$$

The right hand side of the expression is due to the interest towards $\vec{\mathcal{X}}_j^{ff}$. The last expression is a randomization term with $\alpha^{ff}(t)$ being a randomization parameter with $0 \leq \alpha^{ff}(t) \leq 1$ and $\epsilon_i^{ff}(t)$ signifies a vector of several random values selected from a normal or some other distribution during the time, t. The exploitation parameter, $\alpha^{ff}(t)$ can be expressed as:

$$\alpha^{ff}(t) = \beta^{ff}(t)\delta^{ff}(t) \tag{21}$$

Here, $0 < \delta^{ff}(t) < 1$

IV. Framework

The basic architecture of our proposed BIOCAD framework is presented in Fig. 2. Here, the controller module leverages the historical patient vital signs data to train learning models and functions as an optimized classifier and anomaly detector. The WBSN measures patients' vital signs through a plethora of sensors and provides real-time patient data to the controller module. The controller module includes two sub-modules: the classifier and anomaly detector (CAD) module and the bio-inspired optimization (BIO) module.

The CAD module processes the vital signs dataset provided by WBSN and chooses SVM or NN as CM, depending on the performance, for classifying diseases for the patients in real-time. On the other hand, the CAD module uses either an OCSVM model or a DBSCAN model for detecting anomalies in the patient vital signs dataset, which can occur due to measurement error or intentional data alteration. The SVM has two hyper-parameters: C and γ. For each misclassified data point, the C parameter adds a penalty. When C is minimal, the penalty for misclassified points is tiny, so a large-margin decision boundary is selected at the cost of a higher number of misclassifications. The γ parameter regulates the influence distance of a single training point. A low γ value means a strong similarity radius, implying that more points are clustered together. For high γ values, the points must be very similar to each other to be considered as the same category. As a consequence, models with high γ values are vulnerable to overfitting.

Two key hyperparameters that influence the performance of the NN model are the number of hidden layers and the number of nodes in each of the hidden layers. A large value

for the number of hidden layers is appropriate for complex environments, where the model will extract increasingly high-level representations from the data. The inconsistency of the number of hidden layers with the complexity of the problem makes the network overfitted for the training data. Moreover, the number of nodes in the hidden layers must be in the range of input layer size and output layer size.

Two important hyperparameters for OCSVM are: ν and γ. The regularization coefficient is represented by the ν parameter, which controls the upper bound of rejected target data. During the training phase of OCSVM, this parameter is often tuned for rejecting the noise in the target data. Proper choice of ν allows OCSVM to remove distorted training data from the target set correctly, whereas improper ν causes the decision boundary to be skewed by noisy target data or rejects too many target data. The γ parameter in the OCSVM operates exactly similar to the γ parameter in the SVM. The DBSCAN model also has two hyperparameters that have a major impact on the performance of the model. These are ϵ and the minimum number of samples. The ϵ refers to the radius around the core points that defines the cluster boundary. If the radius is too high, the model will develop a deficient number of clusters. For the worst-case scenario, it will develop only one cluster. As a result, it will fail to find the noisy samples as an anomaly detector model. For the minimum number of samples, the higher value will make smaller clusters, and there will be a lot of actual samples that will be considered as noise. So, it will detect a lot of false-positive samples as an anomaly detector.

The BIO module comes into the picture for optimizing all these important hyperparameters that influence both the CMs and ADMs. This module leverages three bio-inspired optimization techniques, the GWO algorithm, the WO algorithm, and the FO algorithm to tune the hyperparameters. The controller module delivers the output from the CAD module to the network controller, which is responsible for detecting anomalies. Besides, the disease classification data is provided to the health care provider center for patient-related decisions and medications. Thus, our proposed framework effectively contributes to both the healthcare and cyber-security domains.

V. Technical Details

In this section, we discuss the technical details of our proposed BIOCAD framework. The overall framework can be divided into three components: classification unit (CU), anomaly detection unit (ADU), and optimization unit (OU). A detailed analysis of the framework components is explained based on Algorithm 1.

A. Classification Unit (CU)

The framework mainly figures out the best ML models and OMs leveraging a grid search on the performance of the available models as shown in Algorithm 1. The CU takes the $dataset$ and splits it into $dataset^{train}$ and $dataset^{test}$. The CMs of the CU are trained on the $dataset^{train}$, and the $dataset^{test}$ is used for determining the performance of the models. Before measuring the performance of the CMs, the hyperparameters of the models are tuned optimally, using the OMs of the OU. The true positive rate (TPR) and true negative rate (TNR) are passed to the $performanceMetric$ function based on the prediction performance of the optimized model on $dataset^{test}$. The $performanceMetric$ function returns the appropriate performance measure depending on the properties of the input dataset (e.g., accuracy or area under the curve metrics are good for the balanced datasets, while for imbalanced datasets, the F1-Score is the most appropriate metric).

B. Anomaly Detection Unit (ADU)

The ADU takes a dataset, $dataset^{anomaly}$, which is generated by adversarial manipulation on the $dataset$. The overall $dataset^{anomaly}$ is solely used for testing the ADMs performance. As the $dataset^{anomaly}$ contains only the anomalous data, all the data samples of this dataset are labeled as an anomaly (-1). However, the ADMs are trained on the $dataset^{train}$ only. The performance of ADMs is measured by correctly identifying the benign and anomalous samples from the $dataset^{benign}$ and $dataset^{anomaly}$. The $dataset^{benign}$ is a slightly modified dataset from the $dataset^{test}$ with altered labels (+1) denoting all benign samples. The hyperparameters of the ADMs are optimized with OU for obtaining the performing model prior to assessing the performance of the models. The $performanceMetric$ function takes the true benign rate (TBR) and true anomaly rate (TAR) based on the prediction performance of the optimized model and returns the appropriate performance measure depending on the properties of the datasets.

C. Optimization Unit (OU)

The OU unit is responsible for optimizing the hyperparameters of all ADMs and CMs. The optimization task is a sequence of exploration and exploitation processes aiming to optimize the hyperparameters based on a fitness function.

Exploration: Exploration refers to a scheme of gathering information about the solution space. It enforces the stochastic behavior to enable dynamic searching of the optimal hyperparameter and avoid being trapped into the local optima. All of our chosen BAs include a specific technique for simulating the exploration behavior. The generic exploration behavior of the BIOCAD framework can be expressed as follows:

$$Exploration_i^{\text{BIOCAD}} = \begin{cases} f(\vec{\mathcal{K}}^{grw}, \vec{\mathcal{C}}^{grw}), & \text{if } i = GWO \\ f(\vec{\mathcal{A}}^{wh}, \vec{\mathcal{D}}^{wh}), & \text{if } i = WO \\ f(\alpha^{ff}), & \text{if } i = FO \end{cases}$$

Exploitation: Exploitation refers to the technique of identifying and utilizing the accumulated information from the exploration behavior. It enforces to converge toward best-observed solutions. The generic exploitation behavior of the BIOCAD framework can be expressed as follows:

$$Exploitation_i^{\text{BIOCAD}} = \begin{cases} f(\vec{-k}), & \text{if } i = GWO \\ f(\vec{\mathcal{D}'}^{wl}, e^{bl}), & \text{if } i = WO \\ f(\beta^{ff}), & \text{if } i = FO \end{cases}$$

Fitness function: The choice of the fitness function is the most crucial part of bio-inspired computing-based optimization. The fitness function

1) Fitness function for CMs: The fitness function of the CMs is the deviation between the predicted and actual labels on the $dataset^{train}$ as denoted by the loss function by the Algorithm 1. As long as the $dataset^{train}$ is trained on enough data to capture the data pattern, the loss function is sufficient to find optimal hyper-parameters for the CMs.

2) Fitness function for ADMs: The fitness function choosing for the ADMs is more challenging as it includes only $dataset^{train}$ with no anomalous samples. Like the CMs, OU uses the loss function as the fitness function. But using only the loss function for the fitness function is not sufficient for all the ADMs. Because in most of the cases, the hyperparameters comes up with flexible boundary around the benign samples of the $dataset^{train}$. However, the flexible boundary performs poorly on the anomalous data samples. Hence, a $tightnessFunction$ is introduced for choosing the fitness function of the ADMs. In the case of OCSVM ADM, we use the sum of the distance of the support vectors from the boundary as the $tightnessFunction$, which overcomes the problem of poor performance on the anomalous samples. For the DBSCAN algorithm, the tightness is ensured by further minimizing the radius of the clusters, eps.

Algorithm 1: Hyperparameter Optimization

Function `GridSearch(`$Models$, $Optimizers$`)`**:**
- $bestModel \leftarrow Null$;
- $bestOptimizer \leftarrow Null$;
- $bestHypParam \leftarrow Null$;
- $bestPerformance \leftarrow 0$;
- **for** *each* $model \in Models$ **do**
  - **for** *each* $optimizer \in Optimizers$ **do**
    - $hypParams \leftarrow$ **Optimization**($model$, $optimizer$)
    - $modelPerformance \leftarrow$ **Performance**($model$, $hypParams$)
    - **if** $modelPerformance > bestPerformance$ **then**
      - $bestModel \leftarrow model$;
      - $bestOptimizer \leftarrow optimizer$;
      - $bestHypParam \leftarrow hyperParams$;
      - $bestPerformance \leftarrow modelPerformance$;
    - **end**
  - **end**
- **end**

return $bestModel, bestHypParam, bestPerformance$**;**

Function `Optimization(`$model$, $optimizer$`)`**:**

$$\underset{hp}{\text{minimize}} \quad \textbf{fitnessFunction}(\text{model}, hp)$$
$$\text{subject to} \quad \forall_{i \in |hp^{model}|} \; lb(hp_i^{model}) <= hp_i^{model} <= ub(hp_i^{model})$$

return hp**;**

Function `Performance(`$model$, $hypParams$`)`**:**
- $mp \leftarrow Null$;
- **if** *type(model) == 'classification'* **then**
  - mp = performanceMetric(TPR,TNR, dataset^{test});
- **end**
- **if** *type(model) == 'anomaly detection'* **then**
  - dataset$^{benign} = createDS(dataset^{test}, label_B)$;
  - mp = performanceMetric(TAR,TBR, (dataset$^{benign} \cup dataset^{anomaly}$));
- **end**

return mp**;**

Function `fitnessFunction(`$model$, $hypParams$`)`**:**
- $fitness \leftarrow Null$;
- $ds \leftarrow$ dataset^{train};
- **if** *type(model) == 'classification'* **then**
  - $fitness$ = loss($model$, $hypParams$, ds);
- **end**
- **if** *type(model) == 'anomaly detection'* **then**
  - $fitness$ = loss($model$, $hypParams$, ds) + tightnessFactor($model$, $hypParams$, ds);
- **end**

return $fitness$**;**

VI. Experiments and Results

This section provides the experimentation to analyze the performance of three state-of-the-art bio-inspired optimization algorithms for medical data CMs and ADMs.

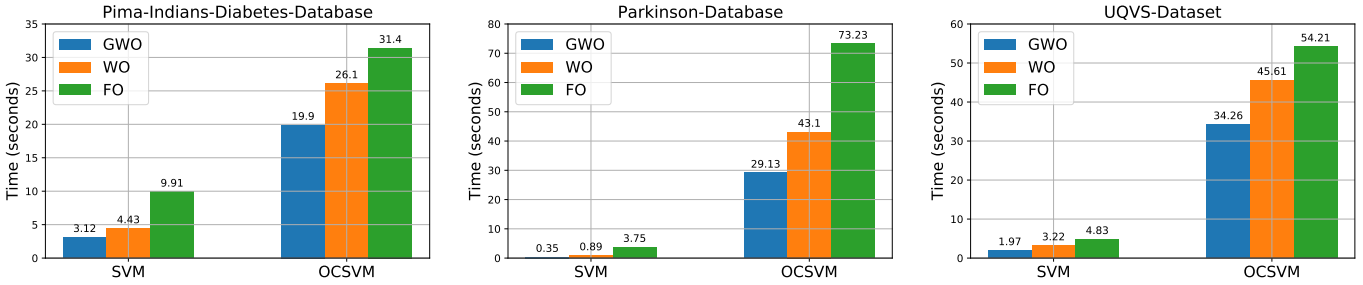
A. Dataset Description

We experiment with three datasets- the Pima Indians diabetes dataset, the Parkinson dataset, and the University of Queensland vital signs (UQVS) dataset for analyzing the performance of the proposed BIOCAD framework. The Pima Indians dataset contains several medical predictor variables (patients' age, insulin level, BMI, number of pregnancies, etc.) and one target variable (diabetes status). The Parkinson dataset is composed of a range of biomedical voice measurements from 31 people, 23 with Parkinson's disease (PD). The UQVS dataset records anesthetic patient monitoring data in 32 cases (3 spinal anesthetics, 25 general anesthetics, 4 sedations) ranging in duration from 13 minutes to 5 hours (median 105 minutes), and we have experimented with a portion of the massive dataset.

In addition, we generate three more datasets from the available dataset for mimicking attack scenarios. For this experimentation purpose, the attack datasets are generated from a slightly deviated data distribution from the normal datasets with the intent to evaluate the OCSVM-based ADMs' performance.

B. Preprocessing

The datasets are split into 75%-25% train-test split before model training, although the attack datasets are solely used for testing purposes. The datasets are further preprocessed by truncating the records with a missing value. For feature selection, the correlation between features is used to remove unnecessary features from the datasets. After removing unnecessary samples and features, the UQVS dataset contains 3000 samples with 25 features, while the Pima Indians Dia-



(a) (b) (c)

Fig. 3. Scalability of the GWO, WO and FO algorithms in optimizing the hyperparameters of SVM and OCSVM on (a) Pima-Indians-Diabetes-Database, (b) Parkinson-Database, and (c) UQVS-Dataset.

TABLE I
CONVERGENCE COMPARISON OF VARIOUS OPTIMIZATION ALGORITHMS FOR SVM AND OCSVM HYPERPARAMETER OPTIMIZATION.

Dataset	Machine Learning Algorithm	Optimization Algorithm	Convergence Epochs
UQVS	SVM	GWO	13
		WO	19
		FO	25
	OCSVM	GWO	41
		WO	69
		FO	71
Parkinson	SVM	GWO	5
		WO	7
		FO	8
	OCSVM	GWO	13
		WO	15
		FO	19
Pima Indians Diabetes	SVM	GWO	9
		WO	16
		FO	14
	OCSVM	GWO	24
		WO	31
		FO	37

betes dataset comprises 795 samples with 8 features, and the Parkinson dataset has got 300 samples with 22 features.

C. Performance Analysis

The performance of the three optimization algorithms of the BIOCAD framework is evaluated based on 4 performance metrics- accuracy, precision, recall, and F1-score. Table II shows the performance of the hyperparameter optimization algorithms on the CMs, while Table III demonstrates the performance comparison for the ADMs.

1) Performance Analysis of BIOCAD framework for the CMs: From Table II, it is evident that on the Pima Indians Diabetes and the UQVS datasets, all three OMs perform reasonably well on SVM and NN-based CMs (except for GWO on Pima Indians dataset for SVM-based CM) in terms of all 4 considered performance metric. However, none of the models demonstrate good performance on the Parkinson dataset. The high false-negative rate is mainly responsible for this performance degradation, as indicated by the low recall value. Increasing the number of agents and number of iterations didn't help them to prevent getting trapped into local optima. All of the models perform significantly well for false-positive cases, which is proven by the perfect precision scores. As the datasets are imbalanced, the F1-score is the best metric for comparing the performance. Based on the performance analysis of both SVM and NN models, it seems that WO outperforms the GWO and FO in most cases.

2) Performance Analysis of BIOCAD framework for the Anomaly Detection Model: The ADM's performance is evaluated on both benign and anomalous data. Although correctly identifying the anomalies is the primary task of ADM, it is not desirable to experience a lot of false alarms. From Table III, it can be observed that on the Parkinson and the UQVS datasets, all three OMs perform significantly well on both ADMs (OCSVM and DBSCAN) in terms of all 4 considered performance metric. However, none of the OMs demonstrate good performance on the Pima Indians Diabetes dataset. The high false benign rate is mainly responsible for this performance degradation, as indicated by the low recall value. Like CMs, in the case of ADMs, WO slightly outperform the other considered models.

D. Convergence

Table I shows the comparison between convergence performance of the optimization algorithms of the BIOCAD frameworks for SVM-based CM and OCSVM-based ADM. The results indicate that the GWO shows the best performance in the case of convergence speed. But performance analysis has shown that GWO performs the worst, which suggests that the GWO optimization algorithm is being trapped in the local optima faster than the WO and FO algorithms. The convergence speed of WO is faster than the FO and not too much slower than the GWO algorithm. Hence, based on the performance analysis and the convergence speed, WO seems to be the better choice for both SVM and OCSVM hyperparameter optimization.

E. Scalability

Fig. 3 shows the execution time comparison for the optimization algorithms in different settings for the SVM-based CM and OCSVM-based ADM model. The Figure indicates that although the sizes of the datasets are considerably different in size, for instance, the UQVS dataset is almost ten

TABLE II
PERFORMANCE COMPARISON OF VARIOUS OPTIMIZATION ALGORITHMS FOR HYPERPARAMETER OPTIMIZATION OF CMS.

Classification Model	Dataset	Optimization Algorithm	γ (SVM)/ Neurons (NN)	C (SVM)/ Layers (NN)	Accuracy	Precision	Recall	F1-Score
SVM	UQVS	GWO	0.003	0.002	0.995	1.0	0.995	0.997
		WO	0.44	0.59	0.988	0.998	0.988	0.9924
		FO	0.18	0.12	0.973	1.0	0.973	0.984
	Parkinson	GWO	0.63	0.02	0.74	1.0	0.72	0.84
		WO	0.3	0.77	0.775	1.0	0.775	0.873
		FO	0.28	0.22	0.775	1.0	0.775	0.873
	pima Indians Diabetes	GWO	1.05	2.1	0.71	1.0	0.69	0.82
		WO	0.28	0.94	0.892	1.0	0.832	0.91
		FO	0.247	0.41	0.82	1.0	0.79	0.88
NN	UQVS	GWO	4	8	0.779	0.829	0.779	0.795
		WO	6	12	0.88	0.89	0.88	0.885
		FO	6	10	0.82	0.85	0.87	0.86
	Parkinson	GWO	6	8	0.79	1.0	0.786	0.88
		WO	4	7	0.825	1.0	0.812	0.895
		FO	8	12	0.741	0.98	0.735	0.84
	pima Indians Diabetes	GWO	5	9	0.81	1.0	0.81	0.895
		WO	3	8	0.842	1.0	0.828	0.906
		FO	8	11	0.808	1.0	0.796	0.886

TABLE III
PERFORMANCE COMPARISON OF VARIOUS OPTIMIZATION ALGORITHMS FOR HYPERPARAMETER OPTIMIZATION OF ADMS.

Anomaly Detection Model	Dataset	Optimization Algorithm	γ (OCSVM)/ ϵ (DBSCAN)	ν (OCSVM)/ minPts (DBSCAN)	TAR	TBR	FAR	FBR	Accuracy	Precision	Recall	F1-Score
OCSVM	UQVS	GWO	0.26	0.1	0.71	0.88	0.11	0.289	0.81	0.94	0.81	0.87
		WO	0.04	0.1	0.826	0.94	0.06	0.174	0.85	0.98	0.83	0.89
		FO	0.034	0.23	0.79	0.91	0.09	0.21	0.85	0.89	0.79	0.84
	Parkinson	GWO	0.014	0.4	0.87	0.92	0.08	0.133	0.77	0.79	0.87	0.828
		WO	0.5	0.69	0.938	1.0	0	0.062	0.95	1.0	0.938	0.968
		FO	0.62	0.52	0.93	1.0	0	0.07	0.95	1.0	0.93	0.965
	Pima Indians Diabetes	GWO	0.6	0.128	1.0	0.875	0.124	0	0.79	0.75	1.0	0.857
		WO	0.38	0.13	1.0	1.0	0	0	1.0	1.0	1.0	1.0
		FO	0.42	0.28	0.932	1.0	0	0.68	0.966	1.0	0.932	0.964
DBSCAN	UQVS	GWO	54.7	3	0.76	1.0	0.0	0.24	0.812	1.0	0.787	0.862
		WO	68	2	0.797	1.0	0.0	0.203	0.831	1.0	0.797	0.887
		FO	156	6	0.758	0.994	0.006	0.242	0.797	0.998	0.758	0.862
	Parkinson	GWO	25.8	5	0.825	0.867	0.133	0.174	0.8125	1.0	0.8125	0.896
		WO	191	3	0.949	1.0	0.0	0.051	0.957	1.0	0.949	0.974
		FO	185	5	0.938	1.0	0.0	0.062	0.949	1.0	0.938	0.968
	Pima Indians Dabetes	GWO	34.63	2	0.941	1.0	0.0	0.058	0.936	1.0	0.91	0.953
		WO	134	7	0.949	1.0	0.0	0.051	0.957	1.0	0.949	0.974
		FO	190	10	0.934	1.0	0.0	0.066	0.945	1.0	0.934	0.966

times larger than the others, the execution time varies almost linearly. Again, it is also clear from the figures that the CM takes a lot less time compared to the ADM. Hence, it can be argued that the optimization algorithms are scalable and feasible to implement for large-scale systems.

VII. CONCLUSION

In this work, we presented a comprehensive digital healthcare system framework referred to as BIOCAD with both patient status/disease classification and anomaly detection capability. The framework uses ML-based models for accomplishing the classification and anomaly detection tasks. Additionally, the ML model hyper-parameters are tuned with bio-inspired computing-based optimization algorithms. Our framework is assessed based on the SVM and NN-based CMs and OCSVM and DBSCAN-based ADMs. We found promising performance for the WO algorithm in the SDHS application. We experimented with three state-of-the-art datasets and got a 0.89-1.0 F1-Score for the WO algorithm. The experimentation also demonstrates that the proposed framework is scalable for both classification and anomaly detection tasks. Moreover, the proposed framework shows significantly high performance in reducing the false alarm rate of the ADMs. However, it might be argued that the anomaly detection rates of the optimized models are not showing perfect performance, which might be alarming for a safety-critical system like medical data anomaly detection. Therefore, in the future, we will evaluate the proposed framework on more promising CMs, ADMs, and OMs to obtain an ameliorated model. Moreover, the proposed framework assumes that the underlying datasets contain sufficient data to capture the data patterns, which might not be valid for the emerging domains. Hence we will use the generative adversarial network-based data augmentation technique.

REFERENCES

[1] National health expenditure projections, 2019âĂŞ28: Expected rebound in prices drives rising spending growth. https://www.healthaffairs.org/doi/full/10.1377/hlthaff.2020.00094?journalCode=hlthaff, 2020. Accessed: 2020-10-20.

[2] Min Chen, Wei Li, Yixue Hao, Yongfeng Qian, and Iztok Humar. Edge cognitive computing based smart healthcare system. *Future Generation Computer Systems*, 86:403–411, 2018.

[3] T. Wehbe, V. Mooney, A. Javaid, and O. Inan. A novel physiological features-assisted architecture for rapidly distinguishing health problems from hardware trojan attacks and errors in medical devices. In *IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 106–109, 2017.

[4] Darlene Storm. Medjack: Hackers hijacking medical devices to create backdoors in hospital networks. https://www.computerworld.com/article/2932371/medjack-hackers-hijacking-medical-devices-to-create-backdoors-in-hospital-networks.html, 2015. Accessed: 2020-01-08.

[5] Ahmad Almogren, Irfan Mohiuddin, Ikram Ud Din, Hisham Al Majed, and Nadra Guizani. Ftm-iomt: Fuzzy-based trust management for preventing sybil attacks in internet of medical things. *IEEE Internet of Things Journal*, 2020.

[6] V Bapuji and D Srinivas Reddy. Internet of things interoperability using embedded web technologies. *International Journal of Pure and Applied Mathematics*, 120(6):7321–7331, 2018.

[7] Rashmi V Deshmukh and Kailas K Devadkar. Understanding ddos attack & its effect in cloud environment. *Procedia Computer Science*, 49:202–210, 2015.

[8] Vahab Pournaghshband, Majid Sarrafzadeh, and Peter Reiher. Securing legacy mobile medical devices. In *International Conference on Wireless Mobile Communication and Healthcare*, pages 163–172. Springer, 2012.

[9] Becky Bracken. Cyberattacks on healthcare spike 45% since november. https://threatpost.com/cyberattacks-healthcare-spike-ransomware/162770/, 2021. Accessed: 2021-03-16.

[10] Leila Hawkins. Cyberattacks increase in healthcare, but sector unprepared. https://www.healthcareglobal.com/technology-and-ai-3/cyberattacks-increase-healthcare-sector-unprepared, 2021. Accessed: 2021-03-16.

[11] Laura Dyrda. The 5 most significant cyberattacks in healthcare for 2020. https://www.beckershospitalreview.com/cybersecurity/the-5-most-significant-cyberattacks-in-healthcare-for-2020.html, 2020. Accessed: 2021-03-16.

[12] Alvi Ataur Khalil, Alexander J Byrne, Mohammad Ashiqur Rahman, and Mohammad Hossein Manshaei. Efficient uav trajectory-planning using economic reinforcement learning. *arXiv preprint arXiv:2103.02676*, 2021.

[13] What does stochastic mean in machine learning? https://machinelearningmastery.com/stochastic-in-machine-learning/, 2020. Accessed: 2021-02-13.

[14] S Binitha, S Siva Sathya, et al. A survey of bio inspired optimization algorithms. *International journal of soft computing and engineering*, 2(2):137–151, 2012.

[15] Nur Imtiazul Haque, Mohammad Ashiqur Rahman, Md Hasan Shahriar, Alvi Ataur Khalil, and Selcuk Uluagac. A novel framework for threat analysis of machine learning-based smart healthcare systems. *arXiv preprint arXiv:2103.03472*, 2021.

[16] AKM Newaz, Nur Imtiazul Haque, Amit Kumar Sikder, Mohammad Ashiqur Rahman, and A Selcuk Uluagac. Adversarial attacks to machine learning-based smart healthcare systems. *arXiv preprint arXiv:2010.03671*, 2020.

[17] Md Hasan Shahriar, Nur Imtiazul Haque, Mohammad Ashiqur Rahman, and Miguel Alonso. G-ids: Generative adversarial networks assisted intrusion detection system. In *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 376–385. IEEE, 2020.

[18] Alvi Ataur Khalil, Javier Franco, Imtiaz Parvez, Selcuk Uluagac, and Mohammad Ashiqur Rahman. A literature review on blockchain-enabled security and operation of cyber-physical systems. *arXiv preprint arXiv:2107.07916*, 2021.

[19] Shubhra Dwivedi, Manu Vardhan, and Sarsij Tripathi. Building an efficient intrusion detection system using grasshopper optimization algorithm for anomaly detection. *Cluster Computing*, pages 1–20, 2021.

[20] Hamid Falaghi, Mahmood-Reza Haghifam, and Chanan Singh. Ant colony optimization-based method for placement of sectionalizing switches in distribution networks using a fuzzy multiobjective approach. *IEEE Transactions on Power Delivery*, 24(1):268–276, 2008.

[21] Liming Shen, Huiling Chen, Zhe Yu, Wenchang Kang, Bingyu Zhang, Huaizhong Li, Bo Yang, and Dayou Liu. Evolving support vector machines using fruit fly optimization for medical data classification. *Knowledge-Based Systems*, 96:61–75, 2016.

[22] Fei Ye. Evolving the svm model based on a hybrid method using swarm optimization techniques in combination with a genetic algorithm for medical diagnosis. *Multimedia Tools and Applications*, 77(3):3889–3918, 2018.

[23] Mingjing Wang and Huiling Chen. Chaotic multi-swarm whale optimizer boosted support vector machine for medical diagnosis. *Applied Soft Computing*, 88:105946, 2020.

[24] Amaal Al Shorman, Hossam Faris, and Ibrahim Aljarah. Unsupervised intelligent system based on one class support vector machine and grey wolf optimization for iot botnet detection. *Journal of Ambient Intelligence and Humanized Computing*, 11(7):2809–2825, 2020.

[25] Sudheer Ch, SK Sohani, Deepak Kumar, Anushree Malik, BR Chahar, AK Nema, Bijaya K Panigrahi, and Ramesh C Dhiman. A support vector machine-firefly algorithm based forecasting model to determine malaria transmission. *Neurocomputing*, 129:279–288, 2014.

[26] John C Platt, John Shawe-Taylor, Alex J Smola, Robert C Williamson, et al. Estimating the support of a high-dimensional distribution. *Technical Report MSR-T R-99–87, Microsoft Research (MSR)*, 1999.

[27] Mete Çelik, Filiz Dadaşer-Çelik, and Ahmet Şakir Dokuz. Anomaly detection in temperature data using dbscan algorithm. In *2011 International Symposium on Innovations in Intelligent Systems and Applications*, pages 91–95. IEEE, 2011.

[28] Xiao-Yuan Yang, Jia Liu, Min-Qing Zhang, and Ke Niu. A new multi-class svm algorithm based on one-class svm. In *international conference on Computational Science*, pages 677–684. Springer, 2007.

[29] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8), 2012.

[30] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. Grey wolf optimizer. *Advances in engineering software*, 69:46–61, 2014.

[31] Seyedali Mirjalili and Andrew Lewis. The whale optimization algorithm. *Advances in engineering software*, 95:51–67, 2016.

[32] Jeremy A Goldbogen, Ari S Friedlaender, John Calambokidis, Megan F Mckenna, Malene Simon, and Douglas P Nowacek. Integrative approaches to the study of baleen whale diving behavior, feeding performance, and foraging ecology. *BioScience*, 63(2):90–100, 2013.

[33] Xin-She Yang. Firefly algorithms for multimodal optimization. In *International symposium on stochastic algorithms*, pages 169–178. Springer, 2009.