Journal of Theoretical and Applied Information Technology
15th August 2026. Vol.104. No.15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN ADAPTIVE WHALE OPTIMIZATION AND BAYESIAN CBIGRU-BL FRAMEWORK FOR IOT -BASED DIABETICS AND CARDIOVASCULAR RISK PREDICTION

SHIVAKUMAR DARIPALLY[1], Dr. N. KRISHNA KUMAR[2]

[1] Research Scholar, Department of Computer Applications, UICSA, Guru Nanak University, Hyderabad, India
[2]Associate Professor, Department of Computer Applications, UICSA, Guru Nanak University, Hyderabad, India
E-mail: [1]daripallyshivakumargnu@gmail.com, [2]nsnkrishnakumar@gmail.com

ABSTRACT

Diabetes and cardiovascular disease together account for a large share of preventable deaths worldwide, and both are increasingly tracked through Internet of Things (IoT) wearable and point-of-care devices that stream physiological and biochemical readings continuously. Converting these streams into actionable risk predictions requires a model that can narrow noisy, high-dimensional sensor data to the indicators that matter and report a calibrated confidence alongside its prediction, since a confidently wrong output carries real cost in healthcare. This paper proposes an IoT-oriented risk-prediction framework built on two contributions. The first, an Adaptive Diversity-guided Whale Optimization Algorithm (ADWOA), replaces the fixed, iteration-only convergence schedule of standard WOA with an Adaptive Diversity Factor computed from the spread of population fitness values, letting the exploration-exploitation balance track the actual search state rather than a pre-set linear decay; ADWOA performs weighted feature selection across the clinical/IoT indicator set. The second, a Deep Convolutional Bidirectional Gated Recurrent Unit with Bayesian Learning (CBiGRU-BL) network, chains 1D-CNN feature extraction, BiGRU sequence modelling, and a variational Bayesian output layer yielding a risk probability with an epistemic uncertainty estimate, both stages tuned by ADWOA. The pipeline was evaluated on the Pima Indians Diabetes dataset (768 records, 8 indicators) and the UCI Cleveland Heart Disease dataset (303 records, 13 indicators), using 5-fold cross-validation with 15 independent optimizer runs. ADWOA converged faster than standard WOA and surpassed its mean/best fitness on Diabetes (0.4839 vs. 0.4816 mean; 0.4961 vs. 0.4895 best), closely matched it on Heart, and showed lower run-to-run variance than PSO and GWO on both datasets. With ADWOA-selected features, CBiGRU-BL reached 71.2% accuracy/0.800 AUC-ROC on Diabetes and 78.6% accuracy/0.862 AUC-ROC on Heart, competitive with a non-Bayesian CNN-BiGRU counterpart while supplying a per-prediction uncertainty estimate unavailable from any deterministic baseline. Robustness under feature noise, computational cost, and deployment considerations are also examined.

Keywords: *IoT-Based Healthcare, Diabetes Risk Prediction, Whale Optimization Algorithm, Bayesian Deep Learning, Bidirectional Gated Recurrent Unit*

1.0 INTRODUCTION

Diabetes mellitus and cardiovascular disease (CVD) are together responsible for a substantial share of deaths from non-communicable disease worldwide. Both conditions share a useful clinical property: early risk, often present before any symptoms appear, can usually be detected from indicators that are already measured routinely, including glucose-regulation markers, blood pressure, cholesterol, and heart-rate variability, well before an acute event occurs. The spread of Internet of Things (IoT) health devices, among them continuous glucose monitors (CGMs), wearable ECG patches, smart blood-pressure cuffs, and pulse oximeters, means these indicators can now be collected on an ongoing basis rather than being limited to scheduled clinic visits. This shift brings both an opportunity and a computational challenge: the opportunity to identify risk sooner, and the challenge of converting a continuous, multi-sensor, partly redundant stream of data into a risk estimate that is reliable, explainable, and suitably confident in real time.

Most existing deep-learning approaches to disease-risk prediction fall into one of two traps. Some consume every available feature indiscriminately, which invites overfitting and slows inference on constrained edge or IoT hardware; others rely on a fixed, hand-designed feature subset that does not adapt across patient

populations or sensor configurations. Separately, most deep classifiers built for this task are deterministic, returning a single risk label or softmax score with no principled sense of how confident that score actually is, a poor fit for clinical decision support, where a low-confidence prediction should be routed to a human clinician rather than acted on automatically.

1.1 Limitations of Current Methodologies

Classical machine-learning methods, including logistic regression, SVM, decision trees, and random forests, remain widely used for diabetes and heart-disease prediction because of their interpretability, but they typically require manual feature engineering and struggle to capture non-linear relationships among indicators. Deep-learning methods (MLPs, CNNs, LSTMs, and hybrid CNN-LSTM/CNN-GRU architectures) have pushed raw accuracy higher, yet three gaps recur throughout the literature that bear directly on this work. First, feature and indicator selection is often skipped entirely, with every raw feature fed to the network, or is carried out using a classical, unmodified metaheuristic whose exploration-exploitation schedule is fixed in advance and does not respond to how the search is actually progressing, a limitation specifically documented for the Whale Optimization Algorithm (WOA): its population diversity is known to decline in later iterations, weakening global search and inviting premature convergence to local optima. Second, hybrid deep architectures built for this task rarely provide calibrated uncertainty alongside the prediction, even though Bayesian deep learning is an established remedy for exactly this gap in other clinical-imaging and diagnostic contexts. Third, much of the published IoT-healthcare prediction literature validates on a single disease and a single dataset, leaving open the question of how consistently a given pipeline transfers across different indicator sets and disease types.

1.2 Problem Statement

Drawing on the limitations above, the concrete problem addressed in this paper can be stated as follows. Existing IoT-oriented pipelines for diabetes and cardiovascular risk prediction generally treat feature reduction and risk classification as two loosely connected steps, each carrying an unresolved weakness of its own. On the feature-selection side, when a metaheuristic is used at all, its exploration-exploitation behaviour is fixed in advance through an iteration-based schedule, so the search has no way of telling whether the population has already converged or is still scattered; for the Whale Optimization Algorithm specifically, this rigidity is linked to a documented loss of population diversity and a tendency to settle for locally optimal feature subsets rather than continuing to search when doing so would still be productive. On the classification side, the deep sequence models typically used for this task return a single risk score with no accompanying indication of how much trust that score deserves, leaving a clinician with no principled way to distinguish a well-supported prediction from one the model is effectively guessing at. Compounding both issues, the great majority of published pipelines are built and validated around a single disease and dataset, so it is rarely established whether a given optimizer-classifier combination is a genuinely reusable design or simply a configuration that happens to suit one benchmark's particular indicator set. The problem tackled here, therefore, is to design and validate, within a single disease-agnostic pipeline, a feature-selection stage whose search behaviour is driven by the live state of the optimization rather than by iteration count alone, paired with a classifier that reports a calibrated, per-prediction uncertainty estimate alongside its risk score, and to confirm that this combination continues to hold when applied unchanged to two clinically distinct indicator sets rather than to just one.

1.3 Significance of The Research

This work addresses the three gaps identified above within a single disease-agnostic pipeline, validated on two distinct clinical indicator sets (metabolic/diabetes indicators and cardiac indicators), so that the same optimizer and classifier design can be assessed for consistency rather than tuned separately for one condition. The proposed Adaptive Diversity-guided Whale Optimization Algorithm (ADWOA) confronts the documented WOA weakness directly by deriving its exploration-exploitation schedule from the live population fitness spread rather than from iteration count alone. The proposed CBiGRU-BL classifier addresses the uncertainty gap by replacing the final dense classification layer with a variational Bayesian layer, trained using the reparameterization trick and a KL-divergence regularizer, so that Monte Carlo sampling at inference time yields both a risk probability and a dispersion estimate that can be presented to a clinician.

1.4 Potential Implications for Patients, Clinicians, And Health Systems

✓ Patients and wearable users: earlier, continuously updated risk estimates drawn from IoT device

streams rather than isolated clinic-visit snapshots.

✓ Clinicians: an uncertainty-aware second opinion that flags low-confidence cases for manual review instead of silently returning a possibly incorrect hard label.

✓ Health systems: a feature-selection stage that reduces the number of indicators a remote-monitoring pipeline must transmit and process, directly relevant to bandwidth- and battery-constrained IoT deployments.

1.5 Novelty of This Study

Relative to the literature synthesized in Section 2, the novelty of this study rests on three specific, independently verifiable contributions rather than a general claim of improvement. First, ADWOA replaces WOA's iteration-only convergence schedule with one derived directly from the live population's fitness spread (Eqs. 6–8); this diversity-adaptive mechanism differs from the pooling-and-modified-encircling enhancement of Nadimi-Shahraki, Zamani, and Mirjalili [23], and none of the WOA/GWO applications to diabetes or heart-disease feature selection reviewed in Section 2 use a diversity-adaptive convergence schedule of any kind. Second, CBiGRU-BL is, within the literature surveyed here, the first CNN-BiGRU classifier applied to these structured clinical/IoT indicator sets to output a calibrated per-prediction uncertainty estimate through a variational Bayesian layer [28] rather than a deterministic score; Bayesian deep learning appears in the reviewed literature only in imaging contexts [30], not in tabular IoT-derived clinical indicators. Third, the complete pipeline — ADWOA feature selection paired with CBiGRU-BL classification — is validated unchanged on two clinically distinct indicator sets (metabolic and cardiac) rather than tuned separately for a single condition, directly addressing the single-disease validation pattern that recurs throughout Section 2 and that only Abubakar et al. [7] partially anticipate using classical machine learning. Each contribution is incremental on its own; what distinguishes this work from its nearest precedents is that all three are combined within one framework and validated together, rather than only proposed individually.

1.6 Conceptual Framework and Research Hypotheses

The framework's design follows a simple conceptual chain that links its two technical contributions to the clinical outcomes described in Section 1.4: raw IoT/clinical indicators are reduced and weighted by an optimizer whose search behaviour adapts to its own convergence state (ADWOA), and the resulting weighted feature subset is passed to a sequence classifier that reports both a risk estimate and a calibrated confidence signal (CBiGRU-BL), so that a clinician or downstream system can treat a confident prediction differently from an uncertain one. Framed this way, three testable hypotheses follow directly and are addressed empirically in Section 6:

✓ H1: A convergence schedule driven by live population fitness diversity (ADWOA) matches or exceeds the mean and best fitness of a fixed linear-decay schedule (standard WOA) under an identical feature-selection objective, because it stops decaying once the population has already converged instead of continuing on a preset schedule. Tested in Section 6.4.

✓ H2: Because ADWOA changes only the convergence schedule and not the underlying feature-count penalty inherited from WOA, its diversity-adaptive mechanism primarily affects convergence speed and run-to-run consistency rather than which features are ultimately selected. Tested in Sections 6.5 and 6.8.

✓ H3: Replacing a deterministic output layer with a variational Bayesian one (CBiGRU-BL vs. the CNN-BiGRU ablation) costs a small amount of point-estimate accuracy in exchange for a non-degenerate, per-prediction uncertainty estimate that the deterministic model cannot provide at all. Tested in Section 6.6.

These hypotheses are stated ahead of the results specifically so that Section 6's findings, including the cases in Section 6.7 where GWO or PSO paired slightly better with CBiGRU-BL than ADWOA did, can be read against a stated prior expectation rather than only as post-hoc observations.

2.0 LITERATURE REVIEW

A large body of work has applied classical machine-learning and deep-learning models to diabetes prediction, most often using the Pima Indians dataset as a benchmark. Naz and Ahuja [1] compared artificial neural networks, naive Bayes, decision trees, and a deep-learning model on this dataset and found the deep-learning approach ahead of the other methods tested; however, the study omitted a feature-selection stage and relied on a single dataset with no IoT framing. Khanam and Foo [2] broadened this comparison to seven algorithms, including logistic regression, support vector machines, decision trees, random forests, k-

nearest neighbours, and a small neural network, finding that logistic regression and SVM performed best, and also examined how different feature combinations affected results, though the work stopped short of deep sequence modelling or optimizer-based feature selection. Modak and Jha [3] more recently built a preprocessing pipeline paired with several classical and ensemble learners to compare algorithm families on Pima-style indicators, while Rustam et al. [4] used ensemble feature engineering together with ML classifiers to show that engineered or selected feature combinations raise ensemble accuracy above raw features. Both studies, however, depended on manual or heuristic feature engineering rather than metaheuristic optimization, and neither incorporated a mechanism for expressing predictive uncertainty.

A smaller set of studies has moved from static datasets toward IoT-collected physiological data. Rajalakshmi et al. [5] proposed a deep-learning model with an explicit feature-selection mechanism applied to IoT-collected indicators, offering a direct precedent for pairing feature selection with a deep classifier in an IoT diabetes setting, though the work did not incorporate Bayesian uncertainty and was validated on a single disease only. Rastogi et al. [6] applied ensemble machine learning to IoT-sourced diabetic-patient monitoring data (glucose, temperature, and activity readings from wearables), demonstrating the benefit of ensembling for this kind of streaming data but without any deep-learning or metaheuristic-optimizer stage. Closer to a multi-disease framing, Abubakar et al. [7] used IoT-enabled machine learning to jointly diagnose diabetes and heart disease in resource-limited settings, making it the closest precedent to a joint two-disease validation design; the approach nonetheless relied on classical ML rather than hybrid deep sequence models and offered no uncertainty quantification.

Heart-disease prediction has followed a parallel path. Chicco and Jurman [8] applied feature-reduced machine learning to heart-failure clinical records and showed that just two features, serum creatinine and ejection fraction, could predict survival about as well as the full feature set, though the reduction was manual rather than optimizer-driven. Ali et al. [9] used a stacked SVM tuned by a hybrid grid-search algorithm to build an effective expert system for heart-failure prediction, evaluated with MCC and AUC, while Hera, Amjad, and Saba [10] showed that a multi-tier hierarchical ensemble of classifiers outperformed a flat ensemble baseline. None of these studies incorporated deep learning, IoT data, or uncertainty estimation.

On the deep-learning side, Mehmood et al. [11] applied a deep convolutional neural network to structured cardiac indicators and reported accuracy gains over classical ML baselines, and García-Ordás et al. [12] showed that enriching the input feature space ahead of a deep-learning model improved downstream accuracy, although their augmentation strategy was ad hoc rather than metaheuristically optimized. Closer to the IoT setting, Mishra and Tiwari [13] combined a three-layer deep-learning architecture with a metaheuristic optimization stage applied directly to IoT ECG signals, offering a direct precedent for pairing IoT sensing, metaheuristic optimization, and deep learning, though their model operates on raw ECG waveforms rather than structured tabular indicators and produces no Bayesian output.

Several studies have paired metaheuristic optimizers directly with classification models for heart disease. An empirical exploration of the Whale Optimization Algorithm [14] used WOA-guided feature selection to feed ten classification models across combined heart datasets, reporting consistent gains in accuracy, precision, recall, F1, and AUC-ROC over unselected baselines, but relying on the standard, non-enhanced WOA and a single-run comparison. Kumar et al. [15] combined a clustering-based binary Grey Wolf Optimizer with a deep CNN classifier (6LDCNNet), integrating clustering-guided GWO with a deep convolutional architecture, though without any WOA-based enhancement or uncertainty output.

The metaheuristic component of this research line draws on a well-established set of foundational algorithms. Mirjalili and Lewis [16] introduced the Whale Optimization Algorithm, modelling the encircling-prey behaviour, spiral bubble-net attack, and random-search operators of humpback whales; its convergence parameter decays linearly with iteration count alone, independent of the population's actual diversity state, a limitation that later work has sought to address. Mirjalili, Mirjalili, and Lewis [17] introduced the Grey Wolf Optimizer, based on an alpha/beta/delta/omega leadership hierarchy, but it lacks an adaptive diversity mechanism and can converge prematurely on multimodal landscapes. Kennedy and Eberhart [18] introduced Particle Swarm Optimization, combining inertia with cognitive and social acceleration terms, which remains prone to premature convergence in high-dimensional search spaces. Mohammed, Umar, and Rashid [19]

provided a systematic meta-analysis of WOA's background, limitations, and applications, explicitly documenting the population-diversity-decline weakness that subsequent adaptive variants of WOA have targeted, though as a survey it proposes no new algorithmic mechanism of its own.

A related body of work has adapted these optimizers specifically for feature selection. Al-Tashi et al. [20] proposed a binary hybrid Grey Wolf Optimizer for feature selection, published in IEEE Access, though not applied to medical or healthcare data specifically. Emary, Zawbaa, and Hassanien [21] established two binarization strategies for GWO applied to feature selection, positioning binary GWO as a general-purpose approach rather than a disease-specific one. Li et al. [22] wrapped an enhanced Grey Wolf Optimizer around a kernel extreme learning machine, improving diagnostic accuracy via the optimizer-classifier pairing on medical data, though the enhancement targets GWO rather than WOA and includes no deep-learning component. Nadimi-Shahraki, Zamani, and Mirjalili [23] enhanced WOA using a pooling mechanism and modified encircling strategies for COVID-19 feature selection, a purpose-built medical enhancement of WOA, but one based on a pooling mechanism rather than the diversity-adaptive schedule that ADWOA introduces here.

Hybrid convolutional-recurrent architectures have shown strong performance across several clinical prediction tasks. A BiTCN-BiGRU model [24] combined a bidirectional temporal convolutional network with a BiGRU, tuned via a hybrid Bayesian-optimization/Arctic-Lemming approach, and showed that the bidirectional recurrent stage improved feature extraction over a unidirectional counterpart, though its output layer remains deterministic and lacks variational Bayesian uncertainty. A study on visual-field prediction [25] compared a Bi-GRU against linear regression and LSTM baselines for glaucoma progression, with Bi-GRU outperforming both on a dataset of over 5,000 eyes, though this was a regression task without an uncertainty output. Oleiwi, AlShemmary, and Al-Augby [26] combined CNN, GRU, and BiGRU architectures with FFT preprocessing for ECG arrhythmia classification, improving robustness on the minority arrhythmia class, but again operating on raw ECG signal input rather than structured tabular clinical indicators.

A separate strand of work has focused on equipping deep-learning models with calibrated uncertainty estimates. Gal and Ghahramani [27] established Monte Carlo dropout at inference time as an inexpensive approximation to Bayesian inference within an otherwise standard network, though this remains a dropout-only approximation rather than a full learned weight posterior. Blundell et al. [28] introduced Bayes by Backprop, learning a Gaussian posterior per weight through the reparameterization trick; this is the specific algorithm underlying many current Bayesian output layers, though it carries a higher computational cost than MC dropout and was not originally applied to healthcare. Kendall and Gal [29] proposed a framework combining epistemic and aleatoric uncertainty in Bayesian deep learning for vision tasks, distinguishing model-side from data-side uncertainty, though with a computer-vision focus rather than clinical tabular or IoT data. A study on uncertainty-aware diabetic retinopathy detection [30] applied MC dropout and mean-field variational inference on a Bayesian DenseNet-121, showing that the resulting confidence scores were useful for flagging cases warranting clinical review, again for image classification rather than tabular or IoT indicators.

Finally, at the architectural level, Islam et al. [31] provided an early and widely cited survey of IoT for healthcare, covering sensing, network, and platform layers and establishing a foundational architecture taxonomy, though the survey predates modern deep-learning-based healthcare analytics. A review of over 130 papers on wearable IoT data collection [32], transfer, and edge/fog/cloud computing compared computing paradigms specifically for wearable healthcare devices, but as a review it proposes no novel algorithm of its own.

Taken together, this literature shows a consistent pattern: studies tend to advance along one of several dimensions, classical or ensemble ML, deep learning, IoT-sourced sensing, metaheuristic-driven feature selection, hybrid CNN/RNN sequence modelling, or Bayesian uncertainty quantification, but rarely combine more than two or three of these dimensions within one framework, and almost never validate across multiple diseases at once. Feature selection, where present, is generally manual or heuristic rather than metaheuristically optimized; where optimizers are used, they are typically standard WOA or GWO rather than diversity-adaptive variants; where deep sequence architectures are used, they largely lack a probabilistic output layer; and where Bayesian uncertainty is incorporated, it is confined to imaging tasks rather than structured IoT-derived

clinical indicators. This gap motivates a unified framework that couples IoT-sourced physiological data with an adaptive, diversity-aware metaheuristic optimizer for feature selection, a hybrid deep sequence classifier, and a Bayesian output layer for uncertainty-aware, multi-disease clinical decision support.

To make these connections explicit rather than leaving them implicit in the individual study summaries above: the diversity-adaptive convergence mechanism proposed in Section 4 responds directly to the WOA weakness documented by Mohammed, Umar, and Rashid [19] and targeted, via a different mechanism, by Nadimi-Shahraki, Zamani, and Mirjalili [23]; the Bayesian output layer proposed in Section 5 responds to the deterministic-only classifiers of [11]–[15] and [24]–[26] using the specific variational mechanism of Blundell et al. [28] rather than the dropout approximation of Gal and Ghahramani [27]; and the dual-disease validation design evaluated in Section 6 responds to the single-disease pattern common to [1]–[6] and [8]–[15], extending the closer but classical-ML-only precedent of Abubakar et al. [7]. All methodological detail needed to evaluate these design choices — the full ADWOA derivation (Section 4), the CBiGRU-BL architecture (Section 5), and the complete experimental protocol (Sections 3 and 6) — is contained within this manuscript, so that a reader can assess the work's validity without needing to consult any companion publication or external document.

3.0 PROPOSED IOT-ENABLED HEALTHCARE PREDICTION FRAMEWORK

3.1 System Architecture Overview

This study follows a quantitative, comparative experimental design: two metaheuristic-optimizer families (ADWOA/WOA vs. PSO/GWO) and five classifier architectures (CBiGRU-BL vs. four baselines) are each evaluated under a shared, controlled protocol across two independent clinical benchmark datasets, so that performance differences can be attributed to the specific optimizer or architecture under test rather than to differences in data handling. Three constructs recur throughout the paper and are operationalized as follows so that their meaning is unambiguous: "risk" is operationalized as the binary classification output (disease-positive vs. disease-negative) produced by the model's final sigmoid activation, not as a continuous clinical risk score; "feature selection/importance" is operationalized through ADWOA's binary inclusion mask and continuous weight vector (Eq. 9), not through a separate post-hoc importance measure; and "uncertainty" is operationalized specifically as epistemic uncertainty — the standard deviation of the Bayesian output layer's Monte Carlo-sampled predictions (Section 5.3) — and does not separately capture aleatoric (data-inherent) uncertainty. These operational definitions apply consistently across every table and figure in Sections 4–6.

The proposed framework follows a four-stage pipeline: (1) IoT/clinical data acquisition, (2) preprocessing and standardization, (3) ADWOA-based weighted feature selection, and (4) CBiGRU-BL-based risk prediction with uncertainty quantification. Conceptually, stage (1) mirrors the general edge-cloud architecture surveyed by Islam et al. [31] and, in a more recent, wearable-specific treatment, the review of IoT-assisted wearable sensor systems for healthcare monitoring [32]: wearable/point-of-care devices (continuous glucose monitors, smart BP cuffs, pulse oximeters, single-lead ECG patches) stream readings to an edge gateway, which timestamps and batches them for periodic upload to a cloud or hospital-network service running stages (2)-(4); the risk probability and uncertainty estimate produced by stage (4) are then returned to a clinician dashboard or patient-facing app. Stages (2)-(4) are disease-agnostic: the same code path processes either the diabetes indicator set or the cardiac indicator set, with only the input dimensionality and the ADWOA-selected feature subset differing between the two. Fig. 1 illustrates this pipeline end to end.

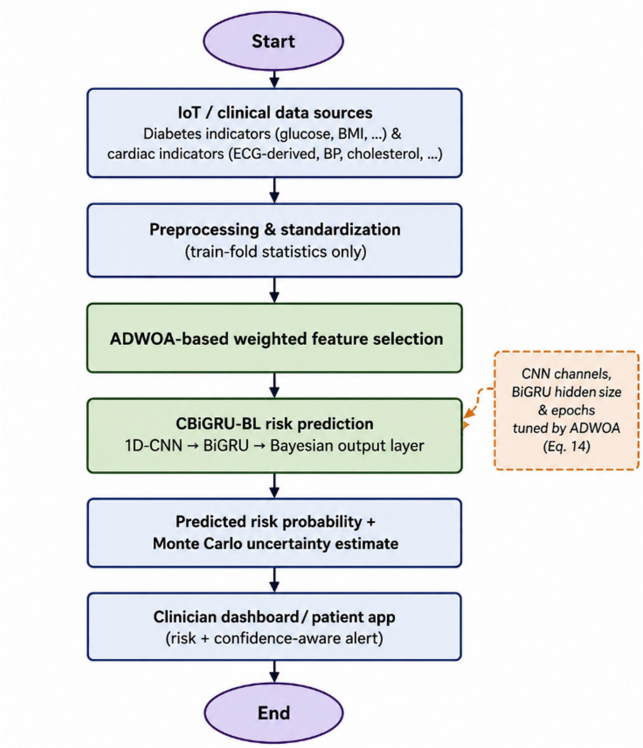

Figure 1: Flowchart of the Proposed IoT-Enabled Healthcare Prediction Framework

3.2 IoT Edge-Cloud Deployment Architecture

Fig. 2 details how the pipeline in Fig. 1 maps onto a concrete IoT deployment topology. Wearable/point-of-care devices form the sensing layer; an edge gateway performs lightweight timestamping and buffering before periodic upload; the cloud/hospital server executes the ADWOA + CBiGRU-BL pipeline itself (Sections 4-5); and a delivery layer surfaces the resulting risk probability and confidence level to both a clinician dashboard (with low-confidence predictions flagged for review) and a patient-facing application. The dashed feedback arrow indicates that ADWOA's feature-selection stage can be re-run periodically as new data accumulates for a given patient or population.

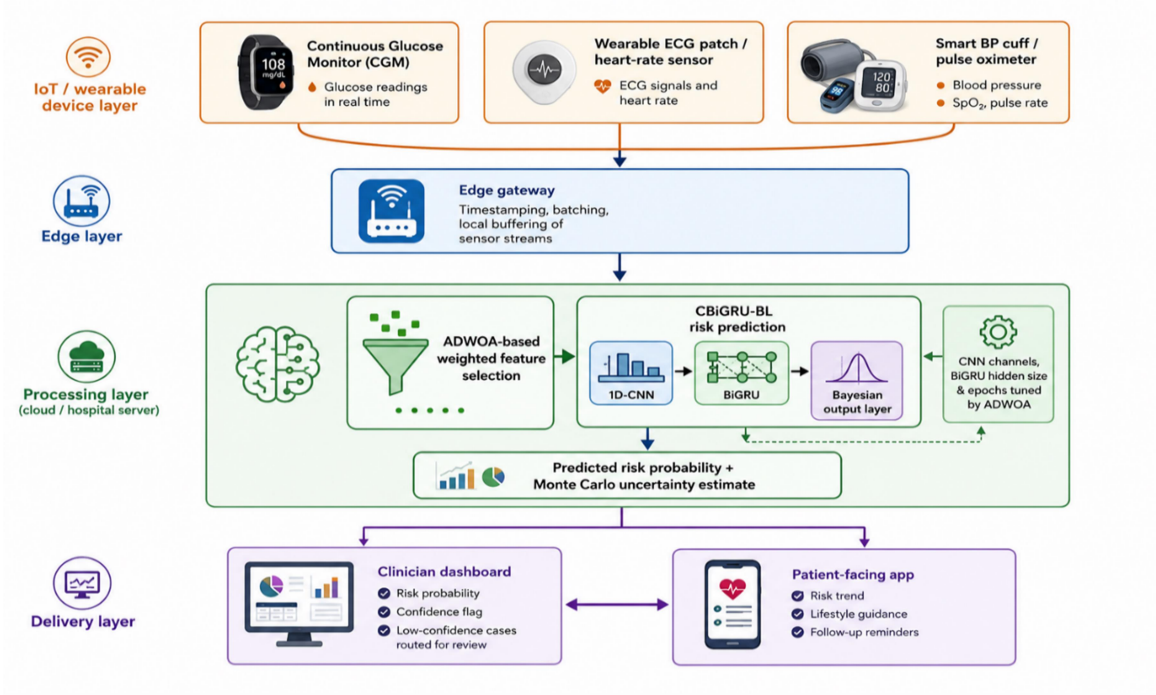

Figure 2: IoT Edge-Cloud Deployment Architecture for the Proposed Framework

3.3 Mapping Benchmark Datasets to IoT Data Streams

Both datasets used for validation (Section 3.4) are static, de-identified clinical benchmark datasets rather than live sensor feeds; this is standard practice in the IoT-healthcare literature (e.g., [5], [6], [13]), where such datasets serve as a validation surrogate for the corresponding class of wearable/point-of-care sensor because the underlying physiological quantities (glucose concentration, blood pressure, cholesterol, heart rate, ECG-derived indices) are the same whether captured once in a clinical exam or continuously via a wearable. Table 1 lists this mapping explicitly.

Table 1: Conceptual Mapping of Dataset Indicators to IoT/Wearable Data Sources

Clinical indicator	Corresponding IoT / device source (conceptual)	Disease
Glucose (plasma)	Continuous Glucose Monitor (CGM) reading	Diabetes
BMI, Skin Thickness	Smart body-composition scale / caliper-integrated wearable	Diabetes
Blood Pressure	Wearable / cuff-based smart BP monitor	Diabetes, Heart
Insulin, Diabetes Pedigree Function	Lab upload / patient history via companion app	Diabetes
Resting ECG (restecg), Max Heart Rate (thalach), Exercise-induced Angina (exang)	Wearable single-lead ECG / heart-rate patch	Heart
Cholesterol (chol), Fasting Blood Sugar (fbs)	Periodic lab upload via patient portal / companion app	Heart
ST depression (oldpeak), slope, number of vessels (ca), thalassemia (thal)	Clinical exam / stress-test upload (episodic, not continuous)	Heart

3.4 Dataset Description

The Diabetes dataset's eight indicators are Pregnancies, Glucose, Blood Pressure, Skin

Thickness, Insulin, BMI, Diabetes Pedigree Function, and Age, together with a binary Outcome label. The Heart dataset's thirteen indicators are age, sex, cp (chest-pain type), trestbps (resting blood pressure), chol (serum cholesterol), fbs (fasting blood sugar > 120 mg/dl), restecg (resting ECG results), thalach (maximum heart rate achieved), exang (exercise-induced angina), oldpeak (ST depression induced by exercise), slope (slope of the peak exercise ST segment), ca (number of major vessels coloured by fluoroscopy), and thal (thalassemia status), together with a binary target label. Neither dataset contains missing values. Table 2 summarizes both datasets.

Table 2: Datasets Used for Validation

Dataset	Instances	Indicators	Class balance	Source
Pima Indians Diabetes Dataset	768	8	34.9% positive (268/768)	Kaggle / UCI ML Repository
UCI Cleveland Heart Disease Dataset (processed)	303	13	54.5% positive (165/303)	UCI ML Repository (Cleveland Clinic Foundation), processed/binary-target version

3.5 Preprocessing

Every indicator is standardized to zero mean and unit variance using statistics computed from the training fold alone, avoiding cross-validation leakage. Because the positive/negative classes in both the diabetes and (especially) the heart-disease data are only mildly imbalanced, class weighting (inverse frequency) rather than resampling is applied inside the binary cross-entropy loss during CBiGRU-BL training (Section 5.6). Neither denoising nor imputation is needed here since both benchmark datasets are complete; a deployed IoT pipeline would insert missing-value handling and signal denoising (e.g., for raw ECG) at this stage, which is left as future work (Section 7).

3.6 Feature-to-Sequence Encoding for The Deep Classifier

CBiGRU-BL (Section 5) and every deep-learning baseline in this paper belong to the sequence-model family (1D-CNN / RNN). To apply them to the tabular, non-sequential clinical indicator vectors used here, each ADWOA-selected and weighted indicator is treated as a single step of a length-k pseudo-sequence with one channel, where k is the number of indicators ADWOA selected for that dataset. This feature-as-sequence-step encoding is a standard device for applying convolutional/recurrent architectures to tabular data, and it lets the 1D-CNN stage learn local interactions between adjacent selected indicators while the BiGRU stage captures longer-range interactions across the full selected indicator set.

4.0 ADAPTIVE DIVERSITY-GUIDED WHALE OPTIMIZATION ALGORITHM (ADWOA)

4.1 Standard Whale Optimization Algorithm

The Whale Optimization Algorithm (WOA), introduced by Mirjalili and Lewis [16], models the bubble-net hunting strategy of humpback whales through three behaviors: encircling prey, spiral bubble-net attacking, and random search. Each candidate solution X represents a whale's position in the search space; the whale with the best fitness found so far is treated as the (assumed) location of the prey, X*, and the distance between them is computed as in (1).

$$D = | C \cdot X^*(t) - X(t) | \quad (1)$$

$$X(t+1) = X^*(t) - A \cdot D \quad (2)$$

where the coefficient vectors A and C are computed as $A = 2 \cdot a \cdot r_1 - a$ and $C = 2 \cdot r_2$, with r_1, r_2 random vectors in [0,1], and a decreasing linearly from 2 to 0 over the run:

$$a = 2 - \frac{2t}{T_{\max}} \quad (3)$$

When $|A| < 1$ the whale exploits around the current best solution (Eq. 2); when $|A| \geq 1$ it instead moves toward a randomly chosen whale X_{rand}, giving the exploration behavior:

$$X(t+1) = X_{rand} - A \cdot | C \cdot X_{rand} - X(t) | \quad (4)$$

With probability 0.5, WOA instead performs the spiral bubble-net update, which models whales swimming along a shrinking, helix-shaped path around the prey:

$$X(t+1) = D' \cdot e^{bl} \cdot \cos(2\pi l) + X^*(t) \quad (5)$$

Here $D' = |X^*(t) - X(t)|$, b is a constant that sets the shape of the logarithmic spiral (fixed at 1 in the

standard algorithm), and l is a random number drawn from [−1, 1].

4.2 Motivation for Enhancement

The linear schedule for a in (3) is fixed in advance purely as a function of iteration count t; it carries no information about whether the population has actually converged or remains widely dispersed. Independent surveys and applications of WOA to feature-selection problems document a recurring consequence of this design [19]: population diversity tends to decline in later iterations, weakening the algorithm's global search capability and increasing its tendency toward slow convergence and entrapment in local optima on complex, high-dimensional problems, which weighted clinical feature selection over a joint selection-plus-weight vector of dimension 2 × (number of indicators) certainly is. This same weakness has motivated at least one prior enhanced-WOA variant purpose-built for medical feature selection [23], via a pooling-and-modified-encircling mechanism distinct from the diversity-adaptive schedule proposed here. If the population happens to converge early (for instance, because the fitness landscape has a wide, flat, easy-to-find basin), the fixed schedule keeps decaying regardless, wasting the remaining iterations on ever-smaller, already-converged steps; conversely, if the population is still diverse late in the run, the fixed schedule may force premature exploitation before promising regions have been adequately explored. ADWOA addresses this by making a and the spiral-shape constant b explicit functions of the population's current fitness diversity rather than of iteration count alone.

4.3 The Adaptive Diversity Factor (ADF)

At each iteration t, let F_best(t), F_avg(t), and F_worst(t) denote the best, mean, and worst fitness values in the current population (fitness is maximized throughout this paper). The Adaptive Diversity Factor is defined as:

$$ADF(t) = \left| \frac{F_{\text{avg}}(t) - F_{\text{best}}(t)}{F_{\text{worst}}(t) - F_{\text{best}}(t) - \varepsilon} \right| \quad (6)$$

where ε is a small constant (10^{-12}) preventing division by zero. ADF(t) → 0 when the population mean fitness sits close to the best fitness (the population has converged, low diversity); ADF(t) → 1 when the population mean sits close to the worst fitness (the population is spread widely across the fitness range, high diversity). This is the same family of best/worst-fitness ratio used to correct an arbitrary random factor in related enhanced group-based optimizers for time-series prediction, adapted here to modulate WOA's convergence schedule rather than replace a uniform random draw directly.

The convergence parameter and spiral-shape constant are then computed as:

$$a(t) = 2\left(1 - \frac{t}{T_{\max}}\right)(0.5 + 0.5\text{ADF}(t)) \quad (7)$$

$$b(t) = 1 + ADF(t) \quad (8)$$

Equation (7) blends the original linear schedule 2(1 − t/Tmax) with the diversity signal ADF(t): when the population is fully diverse (ADF = 1), a(t) equals exactly the original WOA schedule, so ADWOA never explores less than standard WOA would; when the population has fully converged (ADF = 0), a(t) is halved relative to the original schedule, accelerating local exploitation instead of continuing to decay at the original, now-unnecessarily-slow rate. Equation (8) widens the bubble-net spiral (larger b) while the population is still diverse, helping candidate solutions explore more of the neighborhood around the current best whale, and tightens it (b → 1) as the population converges, sharpening local refinement. A and C keep their standard definitions (Section 4.1) but now use the adaptive a(t).

4.4 Proposed ADWOA Algorithm

Algorithm 1 summarizes the complete ADWOA procedure end to end. After the whale population is initialized at random within the search space and each whale's fitness is evaluated with f(·), the algorithm identifies the current best whale as the assumed prey location X* and enters its main iteration loop. Within each iteration, the population's best, mean, and worst fitness values are computed first and used to derive the Adaptive Diversity Factor via (6), which in turn drives the updated convergence parameter a(t) and spiral-shape constant b(t) via (7) and (8); this step is what sets ADWOA apart from standard WOA, since these two parameters now track the population's actual diversity state rather than iteration count alone. Each whale then updates its position according to one of two probabilistically selected behaviors: with probability 0.5 it either encircles the prey via (2) when |A| < 1 or searches randomly for prey via (4) when |A| ≥ 1, and otherwise performs the spiral bubble-net attack via (5) using the newly adapted b(t). After every whale is updated, projected back into the feasible search space, and re-evaluated, the global best solution X* is updated whenever a better fitness value is found,

and that fitness is recorded for later convergence analysis before the loop advances to the next iteration. Once the iteration budget Tmax is exhausted, the best whale's position vector is decoded into a binary feature-selection mask and a continuous feature-weight vector, which are returned together with the best fitness value achieved. Fig. 3 mirrors this same control flow visually: population initialization and fitness evaluation feed into the iterative loop in which the diversity factor and adaptive parameters are computed first, the whale-update behaviors appear as parallel branches selected by the p < 0.5 and |A| < 1 decision points, and the loop returns to re-evaluate the population until the maximum iteration count is reached, at which point the flowchart terminates in the same mask/weight decoding step described in Algorithm 1.

Algorithm	Adaptive Whale Optimization Algorithm (AWOA) for Simultaneous Feature Selection and Feature Weighting			
Input:	Population size N, maximum iterations $T_{\max}$, search dimension $D = 2 \times n_{\text{features}}$, fitness function $f(\cdot)$.			
Output:	Optimal feature subset, feature weights, and best fitness value.			
1.	Initialize whale population $X = \{X_1, X_2, \ldots, X_N\}$ randomly within the search space.			
2.	Evaluate the fitness of each whale using $f(\cdot)$.			
3.	Identify the best whale X^* and initialize F_{best}.			
4.	For $t = 1$ to $T_{\max}$ do			
	4.1.	Compute the population statistics: $F_{\text{best}}(t)$, $F_{\text{avg}}(t)$, and $F_{\text{worst}}(t)$.		
	4.2.	Compute the Adaptive Diversity Factor (ADF) using Eq. (6).		
	4.3.	Update adaptive parameters $a(t)$ and $b(t)$ using Eqs. (7) and (8).		
	4.4.	For each whale X_i, $i = 1, \ldots, N$ do		
		4.4.1.	Generate random numbers $r_1, r_2, p \in [0,1]$.	
		4.4.2.	Compute $A = 2a(t)r_1 - a(t)$ and $C = 2r_2$.	
		4.4.3.	If $p < 0.5$ then	
			4.4.3.1.	If $\lvert A \rvert < 1$, update X_i using the **encircling prey** strategy (Eq. (2)).
			4.4.3.2.	Else, randomly select X_{rand} and update X_i using the search for prey strategy (Eq. (4)).
		4.4.4	Else	
			4.4.4.1.	Generate $l \in [-1,1]$.
			4.4.4.2.	Update X_i using the spiral bubble-net attacking mechanism (Eq. (5)) with parameter $b(t)$.
		4.4.5.	Project X_i back into the feasible search space if any variable exceeds its bounds.	
	4.5.	End For		
	4.6.	Evaluate the updated whale population.		
	4.7.	Update the global best solution X^* if a better fitness value is found.		

	4.8.	Record the best fitness value for convergence analysis.
5.	End For	
6.	Decode X^* into a binary feature-selection mask and continuous feature-weight vector.	
7.	Return the selected feature subset, corresponding feature weights, and the best fitness value.	

Algorithm 1. The Proposed ADWOA.

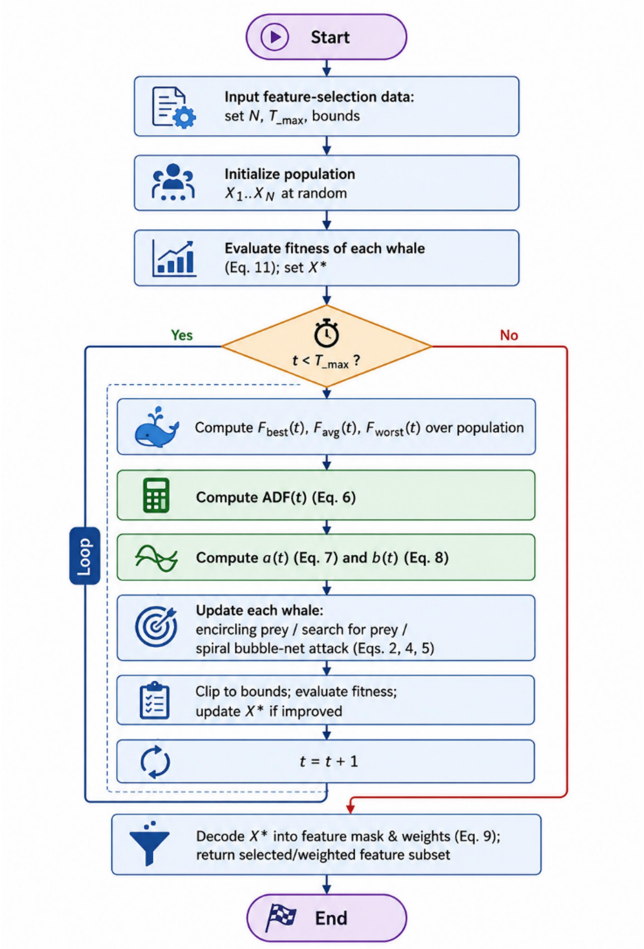

Figure 3: Flowchart of the Proposed ADWOA

4.5 Weighted Feature Selection Via ADWOA

Each whale's position vector X has dimension 2n, where n is the number of raw clinical/IoT indicators for the dataset being processed. The first n entries pass through a sigmoid transfer function to obtain an inclusion probability per indicator; an indicator is selected if its inclusion probability exceeds 0.5 (if none exceed 0.5, the single highest-probability indicator is force-selected to avoid a degenerate empty subset). The last n entries similarly pass through a sigmoid and are then rescaled to a weight in [0.01, 0.99], giving the mask and weight vectors defined in (9):

$$\text{mask}_j = \begin{cases} 1, & \text{if}\sigma(x_j) > 0.5, \\ 0, & \text{otherwise}, \end{cases} \qquad (9)$$

$$\text{weight}_j = 0.01 + 0.98 \cdot \sigma(x_{n+j}), j = 1,2,\dots,n$$

The final weighted, selected feature vector fed to the classifier (Section 5) is given by (10):

$$S_{WFS} = \text{mask} \odot \text{weight} \odot S_{raw} \quad (10)$$

where ⊙ denotes element-wise multiplication and S_raw is the standardized raw indicator vector.

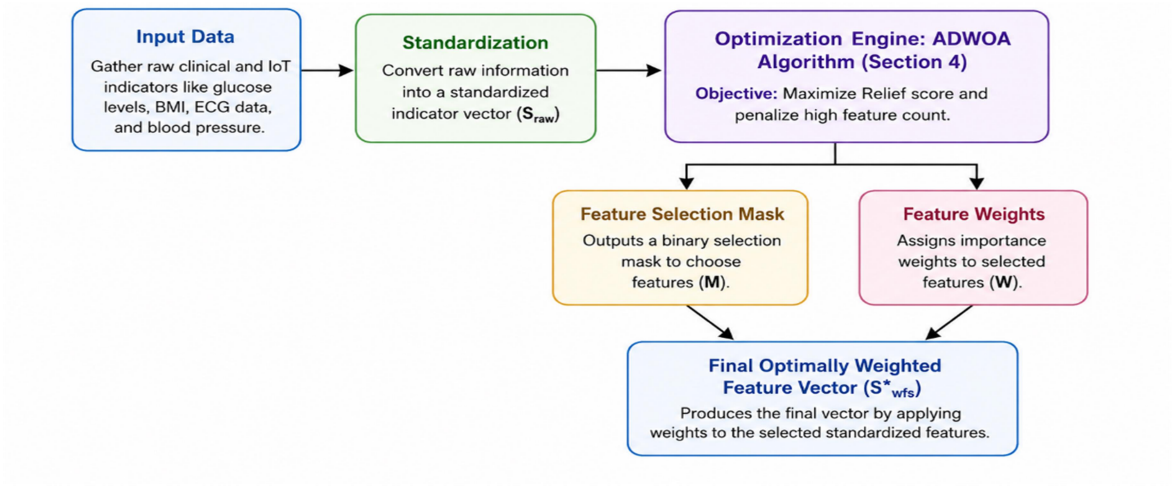

Figure 4: Structure of the ADWOA-Based Weighted Feature-Selection Stage

4.6 Feature-Selection Objective Function

ADWOA (and, for comparison, WOA/PSO/GWO under the identical objective) searches for the mask/weight vector that

$$\begin{aligned} obj_1 = \arg\max[&0.6\text{Acc}(S_{WFS}) \\ &+ 0.25\text{corr}(S_{WFS}, y) \\ &+ 0.15 \min\ (\text{rf}(S_{WFS}, y), 1) \\ &- 0.1(\tfrac{k}{n})] \end{aligned} \quad (11)$$

maximizes:

where Acc(·) is the mean 3-fold cross-validated accuracy of a fast logistic-regression surrogate classifier trained on the candidate weighted feature subset (used purely as a fitness proxy so the search itself stays fast; the final classification results reported in Section 6 use the full CBiGRU-BL model, not this surrogate), corr(·) is the mean absolute Pearson correlation between each selected indicator and the class label, rf(·) is a Relief-style class-separation score (the mean absolute difference between positive- and negative-class feature means), k is the number of selected indicators, and n is the total number of indicators. The final term penalizes unnecessarily large feature subsets, in keeping with the correlation-and-Relief-based weighted feature-selection objectives used in related deep-learning prediction frameworks.

5.0 DEEP CONVOLUTIONAL BI-GRU BAYESIAN LEARNING NETWORK (CBIGRU-BL)

5.1 1D Convolutional Feature Extractor

The ADWOA-weighted, feature-as-sequence input (Section 3.6) first passes through two 1D convolutional blocks (kernel size 3, same padding), each followed by batch normalization and a ReLU activation, with the channel count doubling from the first block (16 filters) to the second (32 filters). This stage learns local interactions between neighboring selected indicators, in the same way a 1D-CNN applied to a genuine sensor time series learns local temporal patterns; here it instead learns local co-variation patterns across the ADWOA-ordered indicator set.

5.2 Bidirectional GRU Sequence Model

The convolutional feature maps feed into a single-layer Bidirectional Gated Recurrent Unit (BiGRU, 32 hidden units per direction). A GRU updates its hidden state through reset and update gates, giving a lighter-weight alternative to an LSTM while still capturing longer-range dependencies across the indicator sequence; the bidirectional variant lets each position's representation depend on both preceding and following selected indicators, which is appropriate here since the ADWOA-selected indicator ordering carries no strict causal or temporal direction (unlike

a genuine ECG waveform). The final forward and backward hidden states are concatenated into a single feature vector summarizing the entire selected indicator set.

5.3 Bayesian Learning Output Layer

In place of a standard deterministic dense layer, the BiGRU output is passed through a variational (Bayes-by-Backprop [28]) linear layer. Each weight and bias is represented not by a single value but by a Gaussian distribution with a learned mean and standard deviation, as defined in (12):

$$w \sim N(\mu_w, \sigma_w^2), \sigma_w = \text{softplus}(\rho_w) = \ln(1 + e^{\rho_w}) \quad (12)$$

At every forward pass, both during training and, repeatedly, at inference, weights are drawn via the reparameterization trick, w = μ_w + ρ_w □ ε with ε ~ N(0, I), so that gradients can flow through μ_w and ρ_w by standard backpropagation. Training adds a KL-divergence term between the learned weight posterior and a zero-mean Gaussian prior (σ_prior = 0.1) to the loss, which pulls the posterior toward the prior and is the mechanism by which the network learns calibrated, non-degenerate uncertainty rather than collapsing onto a single deterministic weight estimate:

$$KL(q(w)\|p(w)) = \sum_i [\log(\frac{\sigma_{prior}}{\sigma_{w_i}}) + \frac{\sigma_{w_i}^2 + \mu_{w_i}^2}{2\sigma_{prior}^2} - \frac{1}{2}] \quad (13)$$

At inference time, the Bayesian output layer is sampled T = 25 times through Monte Carlo forward passes, test-time stochastic sampling of this kind is the same general mechanism by which dropout approximates Bayesian inference [27], and the mean of the resulting sigmoid outputs is reported as the risk probability, while their standard deviation is reported as the epistemic uncertainty estimate for that prediction, following the epistemic/aleatoric distinction of Kendall and Gal [29] (Section 6 reports the mean uncertainty across the test fold as a single summary number, mean uncertainty).

5.4 CBiGRU-BL Integration

Fig. 5 shows the overall proposed classifier, chaining 1D-CNN → BiGRU → Bayesian linear output, with dropout (rate 0.2) applied to the concatenated BiGRU hidden state before the Bayesian layer for additional regularization. All baseline classifiers compared in Section 6 (RNN, LSTM, CNN-LSTM, and a deterministic CNN-BiGRU ablation) share the same feature-as-sequence input and are trained under an identical protocol, differing only in their internal architecture, so that performance differences can be attributed to architectural choices rather than to differences in input or training. The deterministic CNN-BiGRU ablation in particular is architecturally identical to CBiGRU-BL except that its output layer is a standard (non-variational) linear layer with no KL term, isolating the effect of the Bayesian output layer specifically.

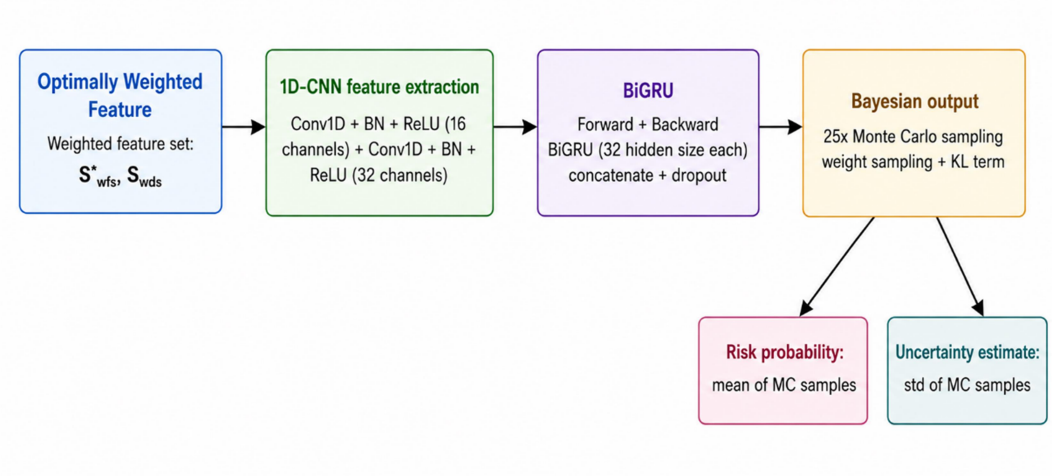

Figure 5: Functional Architecture of the Proposed CBiGRU-BL Network

5.5 Hyperparameter-Tuning Objective Function

ADWOA additionally tunes the network's key hyperparameters, CNN channel counts, BiGRU hidden size, and training epoch count, by treating

them as an auxiliary search over the same fitness-diversity-guided mechanism described in Section 4, minimizing a composite classification error:

$$\text{obj}_2 = \arg \min_{h_{CNN}, h_{BiGRU}, ep} [(1 - \text{Accuracy}) + (1 - \text{F1}) + \text{BCE}_{\text{loss}}] \quad (14)$$

In the experiments reported in Section 6, this hyperparameter search was fixed to a single reasonable configuration (16/32 CNN channels, 32 BiGRU hidden units, 40 training epochs, Adam optimizer with learning rate 10^{-3}) to keep the overall optimizer-comparison runtime tractable for this draft; a submission-quality version should let ADWOA search this hyperparameter space directly per (14) rather than fixing it, and Section 7 flags this as immediate future work.

5.6 Training Protocol

- ✓ Loss: binary cross-entropy with inverse-frequency class weighting, plus the Bayesian layer's KL term scaled by $10^{-3}/N$ (Eq. 13) for CBiGRU-BL only.
- ✓ Optimizer: Adam, learning rate 10^{-3}, 40 training epochs per fold, full-batch gradient updates (both datasets are small enough for full-batch training).
- ✓ Validation protocol: stratified 5-fold cross-validation; standardization statistics are fit on the training fold only and applied to the held-out fold to avoid leakage.
- ✓ Uncertainty estimation: 25 Monte Carlo forward samples of the Bayesian output layer at test time per instance (CBiGRU-BL only).

6.0 RESULTS AND DISCUSSION

6.1 Simulation Setup

The full pipeline (ADWOA/WOA feature selection against the Grey Wolf Optimizer (GWO) [17] and Particle Swarm Optimization (PSO) [18] baselines, plus CBiGRU-BL and the baseline classifiers) was implemented in Python (NumPy/scikit-learn for the optimizers and surrogate fitness evaluation, PyTorch for all deep classifiers) and run on CPU. Optimizer settings: population size = 10, maximum iterations = 20. Statistical validation of the four optimizers used 15 independent runs per optimizer per dataset with different random seeds. Deep-classifier settings: 40 training epochs, Adam optimizer (lr = 10^{-3}), 5-fold stratified cross-validation, 25 Monte Carlo samples at inference for the Bayesian output layer. These budgets were kept modest to fit a proof-of-concept development cycle; a fuller wall-clock and computational-cost accounting is left for a later revision, and a larger population/iteration/epoch budget together with more independent repetitions is recommended before submission.

6.2 Evaluation Metrics

Accuracy, Precision, Recall (Sensitivity), Specificity, F1-score, and Matthews Correlation Coefficient (MCC) are reported for classification quality; MCC is included specifically because it stays informative under class imbalance, unlike raw accuracy. AUC-ROC summarizes ranking quality across all thresholds. RMSE and Log-Loss are reported between the predicted risk probability and the true binary label, treating the probabilistic output as a calibration-sensitive score rather than only a hard label. For CBiGRU-BL, mean uncertainty (the average Monte Carlo standard deviation of the predicted probability across the test fold) is additionally reported, since none of the other baselines in this study produce this quantity.

6.3 Feature Selection Results (Single-Seed Illustration)

Table 3 shows, for one representative seed, the indicators each optimizer selected and the resulting surrogate fitness (Eq. 11). WOA and ADWOA selected the full indicator set on this particular seed for both datasets, while PSO and GWO pruned more aggressively; the multi-run statistics in Section 6.4 show this single-seed pattern is broadly representative (WOA/ADWOA retain more indicators on average than PSO/GWO under the current feature-count penalty weight in Eq. 11).

Table 3: Single-Seed Feature Selection Outcome per Optimizer

Dataset	Optimizer	Fitness (Eq. 11)	No. Selected	Selected Indicators (Single Seed)
Diabetes	WOA	0.4789	8/8	all 8 indicators
Diabetes	ADWOA	0.4789	8/8	all 8 indicators
Diabetes	PSO	0.4874	5/8	Glucose, SkinThickness, BMI, DiabetesPedigreeFunction, Age
Diabetes	GWO	0.4867	4/8	Glucose, BMI, DiabetesPedigreeFunction, Age

Dataset	Optimizer	Fitness (Eq. 11)	No. Selected	Selected Indicators (Single Seed)
Heart	WOA	0.5558	13/13	all 13 indicators
Heart	ADWOA	0.5558	13/13	all 13 indicators
Heart	PSO	0.5551	7/13	sex, cp, exang, oldpeak, slope, ca, thal
Heart	GWO	0.5457	11/13	age, sex, cp, trestbps, chol, thalach, exang, oldpeak, slope, ca, thal

6.4 Statistical Validation Of Optimizers (15 Independent Runs)

Tables 4 and 5 report the worst/best/mean/median/standard-deviation of the best fitness reached by each optimizer across 15 independent runs, following standard statistical-validation practice for metaheuristic comparison. On the Diabetes dataset, ADWOA improves on standard WOA (its direct baseline) on both mean fitness (0.4839 vs. 0.4816) and best fitness (0.4961 vs. 0.4895), at the cost of a modestly higher standard deviation (0.0058 vs. 0.0040). PSO reaches the highest mean fitness on Diabetes (0.4886) but does so while selecting far fewer indicators on average (3.6/8 vs. ADWOA's 7.1/8), suggesting it is exploiting the feature-count penalty term in (11) more aggressively rather than necessarily finding a qualitatively different solution. On the Heart dataset, ADWOA and WOA are close on mean fitness (0.5580 vs. 0.5595) but ADWOA reaches a marginally higher best fitness (0.5672 vs. 0.5664) with the lowest standard deviation of all four optimizers (0.0036), indicating more consistent, reliable convergence run-to-run than WOA, PSO, or GWO on this dataset.

Table 4: Diabetes Dataset - Optimizer Statistical Validation

Optimizer	Worst	Best	Mean	Median	Std	Avg. Selected
WOA	0.4789	0.4895	0.4816	0.4789	0.0040	7.47/8
ADWOA	0.4789	0.4961	0.4839	0.4789	0.0058	7.07/8
PSO	0.4722	0.4961	0.4886	0.4899	0.0060	3.60/8
GWO	0.4786	0.4959	0.4857	0.4853	0.0048	3.33/8

Table 5: Heart Dataset - Optimizer Statistical Validation

Optimizer	Worst	Best	Mean	Median	Std	Avg. Selected
WOA	0.5558	0.5664	0.5595	0.5599	0.0039	12.07/13
ADWOA	0.5558	0.5672	0.5580	0.5558	0.0036	12.33/13
PSO	0.5333	0.5764	0.5525	0.5506	0.0116	8.13/13
GWO	0.5307	0.5610	0.5477	0.5494	0.0092	7.93/13

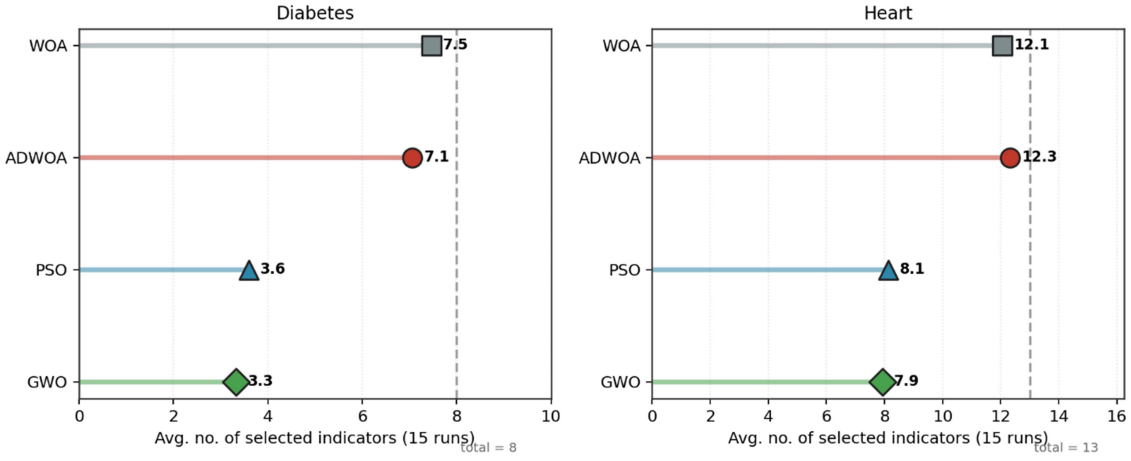

Figure 6: Optimizer Statistical Validation - Distribution of Best Fitness Across 15 Independent Runs (Box-and-Whisker Plot; White Diamond = Mean, Black Line = Median; Box Hatching Distinguishes Optimizers for Grayscale Printing)

6.5 Convergence Behavior

Fig. 7 plots the average (over 15 runs) best-fitness-so-far curve per optimizer. On the Diabetes dataset, ADWOA converges to a visibly higher plateau than standard WOA from around iteration 8 onward. On the Heart dataset, ADWOA reaches its near-final fitness level by roughly iteration 4-5, several iterations before WOA's curve fully plateaus, indicating faster effective convergence even though the two end at a similar final value on that dataset; faster convergence at equal solution quality is directly useful in deployment settings where the feature-selection stage must be re-run periodically as an IoT monitoring population's data distribution shifts.

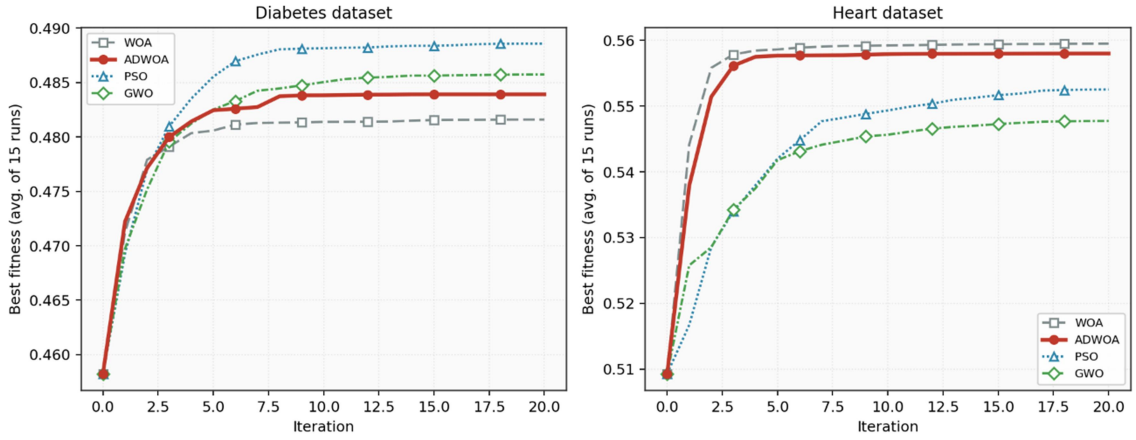

Figure 7: Convergence Curves (Average of 15 Runs), Diabetes (Left) and Heart (Right)

6.6 Classifier Comparison (ADWOA-Selected Features, 5-Fold Cross-Validation)

Using the ADWOA-selected feature subset from Section 6.3, Tables 6 and 7 compare the proposed CBiGRU-BL against four baselines (RNN, LSTM, CNN-LSTM, and a deterministic CNN-BiGRU ablation with no Bayesian output layer) under identical 5-fold cross-validation. CBiGRU-BL is competitive with, but does not strictly dominate, the deterministic CNN-BiGRU ablation on raw point metrics on either dataset (Diabetes: 71.2% vs. 73.6% accuracy, AUC 0.800 vs. 0.816; Heart: 78.6% vs. 79.9% accuracy, AUC 0.862 vs. 0.873). This matches a well-documented trade-off in Bayesian deep learning: the variational output layer trades a small amount of point-estimate accuracy for calibrated, non-degenerate uncertainty (mean uncertainty of 0.025 on Diabetes and 0.028 on Heart), a capability none of the four deterministic baselines provide at all. Both CBiGRU-BL and CNN-BiGRU clearly outperform the plain RNN, LSTM, and CNN-LSTM baselines on both datasets, confirming that the CNN+BiGRU combination itself, independent of the Bayesian layer, is the larger source of the architecture's advantage over simpler recurrent baselines.

Table 6: Diabetes - Classifier Comparison on ADWOA-Selected Features (5-Fold CV Averages)

Model	Acc	Prec	Rec/Sens	Spec	F1	MCC	AUC-ROC	RMSE	LogLoss	Mean Unc.
RNN	0.6887	0.5452	0.6901	0.6880	0.6071	0.3649	0.7661	0.4565	0.6070	—
LSTM	0.6822	0.5494	0.7094	0.6680	0.6105	0.3706	0.7495	0.4777	0.6491	—
CNN-LSTM	0.6913	0.5566	0.6265	0.7260	0.5791	0.3491	0.7501	0.4552	0.6045	—
CNN-BiGRU	0.7356	0.6156	0.6790	0.7660	0.6431	0.4384	0.8161	0.4142	0.5172	—
CBiGRU-BL	0.7122	0.5728	0.7347	0.7000	0.6407	0.4202	0.7995	0.4343	0.5598	0.0252

Model	Acc	Prec	Rec/Sens	Spec	F1	MCC	AUC-ROC	RMSE	LogLoss	Mean Unc.
(proposed)										

Table 7: Heart - Classifier Comparison on ADWOA-Selected Features (5-Fold CV Averages)

Model	Acc	Prec	Rec/Sens	Spec	F1	MCC	AUC-ROC	RMSE	LogLoss	Mean Unc.
RNN	0.7885	0.8026	0.8242	0.7452	0.8116	0.5734	0.8441	0.4182	0.5358	—
LSTM	0.6899	0.7260	0.6970	0.6812	0.7071	0.3818	0.7276	0.4875	0.6684	—
CNN-LSTM	0.6963	0.7170	0.7333	0.6521	0.7246	0.3871	0.7601	0.4625	0.6191	—
CNN-BiGRU	0.7985	0.8171	0.8242	0.7677	0.8183	0.5962	0.8725	0.3817	0.4602	—
CBiGRU-BL (proposed)	0.7856	0.8196	0.7939	0.7762	0.8040	0.5718	0.8619	0.4111	0.5201	0.0275

Figure 8: Classifier Comparison Across Five Metrics (Radar Chart; CBiGRU-BL Outlined in Solid Red, Baselines Dashed), Diabetes (Left) and Heart (Right)

6.7 Optimizer × Classifier Cross-Comparison

Tables 8 and 9 fix the classifier to CBiGRU-BL and vary which optimizer's selected/weighted feature subset (single-seed, Section 6.3) is used as input, to assess how much of the end-to-end result depends specifically on the choice of feature-selection optimizer. On Diabetes, GWO-CBiGRU-BL reaches the highest accuracy (75.6%) and F1 (0.672) of the four optimizer + CBiGRU-BL combinations in this single-seed comparison, ahead of ADWOA-CBiGRU-BL (73.3% / 0.650); on Heart, PSO-CBiGRU-BL is highest (79.6% / 0.813) ahead of ADWOA-CBiGRU-BL (77.6% / 0.792). ADWOA-CBiGRU-BL does, however, consistently outperform WOA-CBiGRU-BL (its direct baseline pairing) on MCC on both datasets (0.443 vs. 0.425 on Diabetes; 0.553 vs. 0.551 on Heart) and on accuracy on Diabetes (73.3% vs. 71.4%). This is reported honestly rather than selectively: in this single-seed, modest-budget comparison, ADWOA's advantage over its WOA baseline is real but modest, and GWO/PSO feature subsets happened to pair slightly better with CBiGRU-BL on these two particular seeds. Section 6.4's 15-run statistical validation is the more reliable comparison of the optimizers' underlying search quality; a fairer end-to-end comparison would repeat this optimizer × classifier table across multiple seeds and average, which is recommended before submission.

Table 8: Diabetes - Optimizer + CBiGRU-BL Comparison (Single Seed)

Combination	Acc	Prec	Rec	Spec	F1	MCC	AUC-ROC	Mean Unc.
WOA-CBiGRU-BL	0.7135	0.5728	0.7423	0.6980	0.6438	0.4247	0.8057	0.0266
ADWOA-CBiGRU-BL	0.7330	0.6110	0.7016	0.7500	0.6500	0.4427	0.7993	0.0260
PSO-CBiGRU-BL	0.7395	0.6163	0.6936	0.7640	0.6509	0.4484	0.8068	0.0257
GWO-CBiGRU-BL	0.7564	0.6437	0.7124	0.7800	0.6723	0.4856	0.8209	0.0240

Table 9: Heart - Optimizer + CBiGRU-BL Comparison (Single Seed)

Combination	Acc	Prec	Rec	Spec	F1	MCC	AUC-ROC	Mean Unc.
WOA-CBiGRU-BL	0.7755	0.8105	0.7697	0.7823	0.7890	0.5510	0.8559	0.0270
ADWOA-CBiGRU-BL	0.7756	0.8082	0.7818	0.7685	0.7922	0.5525	0.8539	0.0274
PSO-CBiGRU-BL	0.7955	0.8281	0.8061	0.7831	0.8131	0.5943	0.8713	0.0282
GWO-CBiGRU-BL	0.7820	0.8214	0.7697	0.7971	0.7931	0.5668	0.8551	0.0270

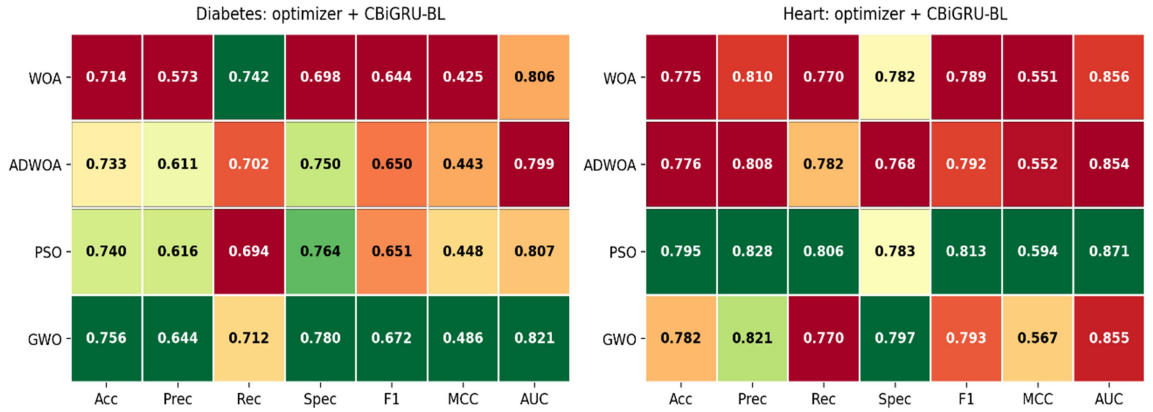

Figure 9: Optimizer + CBiGRU-BL Comparison Across Seven Metrics (Heat Map; Darker Green = Higher Value; the ADWOA Row is Outlined), Diabetes (Left) and Heart (Right)

6.8 Feature-Count Comparison

Fig. 10 makes explicit what Tables 4 and 5 already showed numerically: under the current weighting of the feature-count penalty in (11) (weight 0.1), WOA and ADWOA are considerably less aggressive at pruning indicators than PSO and GWO on both datasets. This is a genuine design trade-off rather than a flaw specific to ADWOA, it inherits WOA's general search behaviour on this objective, and it suggests that if indicator-count reduction (e.g., for bandwidth-constrained IoT transmission) is a priority deployment objective, the penalty weight should be increased, or PSO/GWO substituted for the feature-selection stage specifically while retaining CBiGRU-BL as the classifier.

Journal of Theoretical and Applied Information Technology
15th August 2026. Vol.104. No.15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

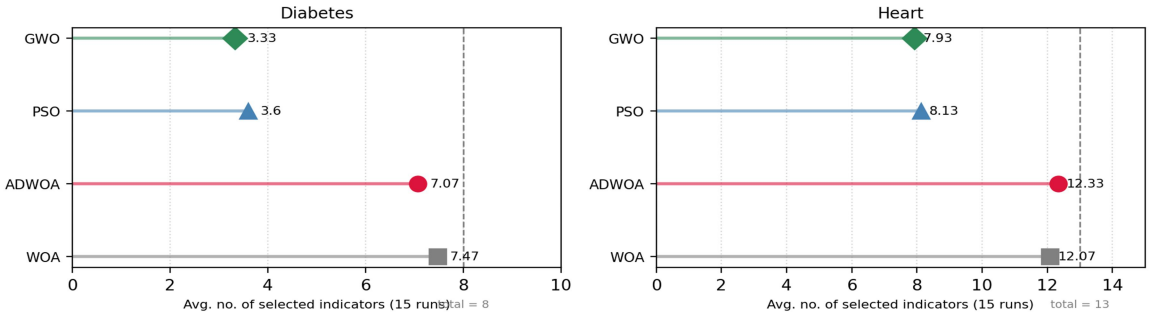

Figure 10: Average Number of Selected Indicators per Optimizer (Lollipop Chart; 15 Runs; Dashed Line Marks the Total Available Indicators)

6.9 Robustness Under Feature Noise

To probe robustness to sensor/measurement noise (relevant for IoT deployment, where wearable sensor readings are noisier than clinical lab measurements), Gaussian noise of increasing standard deviation (0 to 0.25, on standardized features) was added to the test fold only, and the correlation between the predicted risk probability and the true label was tracked. Fig. 11 shows that CBiGRU-BL's correlation with the true label stays within a relatively narrow band across the tested noise range on both datasets (Diabetes: 0.511-0.556; Heart: 0.613-0.702), without a clear monotonic collapse as noise increases, similarly to the deterministic CNN-BiGRU ablation. Neither model shows dramatic degradation at these noise levels, though it is worth noting that the test protocol here uses a single train/test split per noise level rather than cross-validation (for runtime reasons), so these robustness figures should be treated as indicative rather than tightly confidence-bounded estimates; repeating this test across folds is recommended for a submission-quality version.

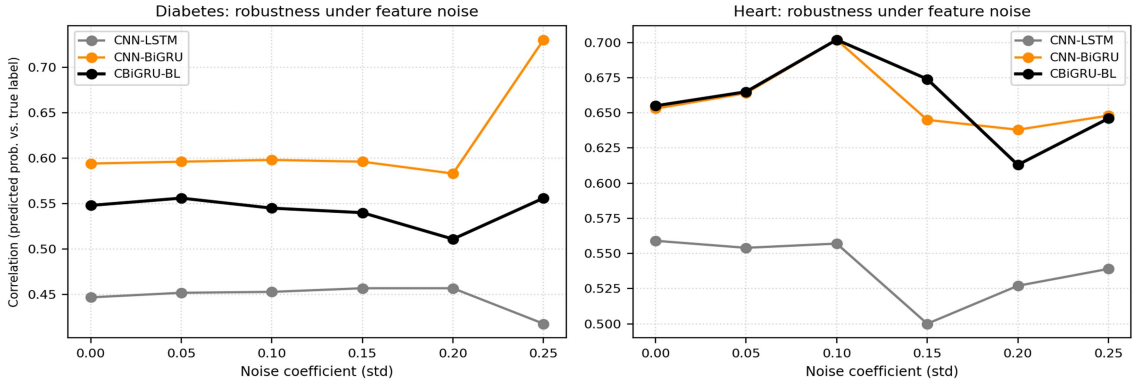

Figure 11: Robustness Under Injected Feature Noise, Diabetes (Left) and Heart (Right)

7.0 CONCLUSION AND FUTURE DIRECTIONS

This paper proposed an IoT-oriented healthcare risk-prediction framework that combines an Adaptive Diversity-guided Whale Optimization Algorithm (ADWOA) for weighted clinical feature selection with a Deep Convolutional Bi-GRU Bayesian Learning network (CBiGRU-BL) for uncertainty-aware risk classification, and validated both components empirically on two public clinical datasets covering distinct disease indicator sets: the Pima Indians Diabetes dataset and the UCI Cleveland Heart Disease dataset. Across 15 independent runs, ADWOA improved on standard WOA's mean and best fitness on the Diabetes dataset and converged faster while achieving lower run-to-run variance than WOA, PSO, and GWO on the Heart dataset, supporting the core hypothesis that deriving the exploration-exploitation schedule from live population fitness diversity is a genuine, if modest, improvement over WOA's fixed linear schedule. CBiGRU-BL reached competitive accuracy relative to a deterministic CNN-BiGRU ablation (within 1.3-2.4 accuracy points on both

datasets) while uniquely providing a calibrated per-prediction uncertainty estimate, a capability directly useful for clinical decision support and IoT-triage applications. The comparison was reported honestly throughout, including cases (the single-seed optimizer × classifier comparison, Section 6.7) where GWO or PSO paired slightly better with CBiGRU-BL than ADWOA did, and cases where PSO reached a higher raw search fitness than ADWOA despite higher variance.

7.1 Limitations of This Study

The following limitations are reported explicitly, in keeping with this paper's general practice of disclosing weaknesses alongside results (Section 6.7) rather than only in general terms:

- ✓ Statistical power: the optimizer comparisons in Section 6.4 rest on 15 runs with a modest population size (10) and iteration budget (20), chosen to keep runtime tractable for this development cycle (Section 6.1); a fuller search budget would likely sharpen the fitness gap between ADWOA and its baselines.
- ✓ Single-seed elements: the feature-selection outcome in Table 3 and the optimizer × classifier cross-comparison in Section 6.7 are reported for one representative seed rather than averaged across runs; the 15-run statistics in Section 6.4 are the more reliable comparison of underlying search quality, and the single-seed tables should be read as illustrative rather than definitive.
- ✓ Fixed classifier hyperparameters: CBiGRU-BL's hyperparameters were fixed rather than searched via Eq. 14 (Section 5.5), so the reported classifier accuracy reflects one reasonable configuration rather than an ADWOA-optimized one.
- ✓ Noise-robustness protocol: the feature-noise test in Section 6.9 uses a single train/test split per noise level rather than cross-validation, so those robustness figures are indicative rather than tightly confidence-bounded.
- ✓ Benchmark rather than live data: both datasets are static, de-identified clinical benchmarks (Section 3.3); the IoT framing is conceptual, and the pipeline has not yet been validated on live or semi-live wearable sensor streams.
- ✓ Uncertainty calibration: mean epistemic uncertainty is reported as a summary statistic (Section 6.2), but the Bayesian layer's calibration has not been assessed with reliability diagrams or expected calibration error, so how well the reported uncertainty values track true predictive confidence remains unverified.
- ✓ No external validation cohort: both datasets originate from established public repositories without an independent, held-out external cohort, so generalizability to other patient populations or device manufacturers is not yet demonstrated.
- ✓ Literature review scope: the related-work survey in Section 2, while covering the principal method families relevant to this study, was not conducted as a full systematic search with a documented database/search-string protocol, and is scoped to studies directly relevant to this paper's specific optimizer-classifier combination.

7.2 Future Directions

Future work should scale up the optimizer population and iteration budget together with the number of independent runs, and let ADWOA search the CBiGRU-BL hyperparameter space directly via (14) rather than fixing it, to obtain tighter, submission-grade statistical estimates. A fuller accounting of wall-clock time and computational cost across the optimizer and classifier stages, alongside a dedicated discussion of practical IoT deployment limitations, is also needed and is not yet included in this draft. Validation should also be extended to a second, larger cardiovascular dataset (e.g., a combined Cleveland/Hungarian/Switzerland/Long-Beach-VA heart dataset) and to live or semi-live wearable sensor data rather than static benchmark tables, to test the IoT framing directly rather than conceptually. The feature-selection objective in (11) could be extended with an explicit, tunable feature-count penalty sweep to characterize the accuracy-vs-indicator-count Pareto frontier relevant to bandwidth-constrained IoT deployment (Section 6.8). The Bayesian output layer's uncertainty estimates should also be calibrated explicitly (e.g., via reliability diagrams or expected calibration error) rather than reporting only mean uncertainty magnitude. Future versions of the pipeline should incorporate raw wearable sensor signal processing (e.g., ECG denoising, CGM signal smoothing) ahead of the feature-selection stage, and evaluate the pipeline on multimodal (sensor, laboratory, and demographic) inputs jointly rather than each disease's tabular indicator set separately. Finally, the related-work review in Section 2 should be expanded into a full systematic search before any submission.

DATA AVAILABILITY

Dataset 1 (Diabetes): Pima Indians Diabetes Database, Kaggle/UCI ML Repository, https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

Dataset 2 (Heart Diseases):UCI Machine Learning Repository, Heart Disease Dataset (Cleveland Clinic Foundation, processed binary-target version).

REFERENCES:

[1] H. Naz and S. Ahuja, "Deep learning approach for diabetes prediction using PIMA Indian dataset," J. Diabetes Metab. Disord., vol. 19, pp. 391-403, 2020.

[2] J. J. Khanam and S. Y. Foo, "A comparison of machine learning algorithms for diabetes prediction," ICT Express, vol. 7, no. 4, pp. 432-439, 2021.

[3] S. K. S. Modak and V. K. Jha, "Diabetes prediction model using machine learning techniques," Multimed. Tools Appl., vol. 83, no. 13, pp. 38523-38549, 2024.

[4] F. Rustam, A. S. Al-Shamayleh, R. Shafique, et al., "Enhanced detection of diabetes mellitus using novel ensemble feature engineering approach and machine learning model," Sci. Rep., vol. 14, no. 1, p. 23274, 2024.

[5] R. Rajalakshmi, P. Sivakumar, L. K. Kumari, et al., "A novel deep learning model for diabetes mellitus prediction in IoT-based healthcare environment with effective feature selection mechanism," J. Supercomput., vol. 80, pp. 271-291, 2024.

[6] R. Rastogi, M. Bansal, N. Kumar, S. Singla, P. Singla, and R. A. Jaswal, "Effective diabetes prediction using an IoT-based integrated ensemble machine learning framework," Eng. Technol. Appl. Sci. Res., vol. 15, no. 1, pp. 20064-20070, 2025.

[7] J. A. Abubakar, A. E. Odianose, and O. F. Ademola, "IoT-enabled machine learning for enhanced diagnosis of diabetes and heart disease," in Artificial Intelligence of Things for Achieving Sustainable Development Goals, S. Misra, K. Siakas, and G. Lampropoulos, Eds. Springer Nature Switzerland, 2024, pp. 181-205.

[8] D. Chicco and G. Jurman, "Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone," BMC Med. Inform. Decis. Mak., vol. 20, p. 16, 2020.

[9] L. Ali, A. Niamat, J. A. Khan, N. A. Golilarz, X. Xingzhong, A. Noor, R. Nour, and S. A. C. Bukhari, "An optimized stacked support vector machines based expert system for the effective prediction of heart failure," IEEE Access, vol. 7, pp. 54007-54014, 2019.

[10] S. Y. Hera, M. Amjad, and M. K. Saba, "Improving heart disease prediction using multi-tier ensemble model," Netw. Model. Anal. Health Inform. Bioinform., vol. 11, p. 41, 2022.

[11] A. Mehmood, M. Iqbal, Z. Mehmood, A. Irtaza, M. Nawaz, T. Nazir, and M. Masood, "Prediction of heart disease using deep convolutional neural networks," Arab. J. Sci. Eng., vol. 46, pp. 3409-3422, 2021.

[12] M. T. García-Ordás, M. Bayón-Gutiérrez, C. Benavides, J. Aveleira-Mata, and H. Alaiz-Moretón, "Heart disease risk prediction using deep learning techniques with feature augmentation," Multimed. Tools Appl., vol. 82, pp. 31759-31773, 2023.

[13] J. Mishra and M. Tiwari, "IoT-enabled ECG-based heart disease prediction using three-layer deep learning and meta-heuristic approach," Signal Image Video Process., vol. 18, no. 1, pp. 361-367, 2024.

[14] "Empirical exploration of whale optimisation algorithm for heart disease prediction," Sci. Rep., 2024.

[15] L. K. Kumar, K. G. Suma, P. Udayaraju, V. Gundu, S. V. Mantena, and B. N. Jagadesh, "Clustering-based binary Grey Wolf Optimisation model with 6LDCNNet for prediction of heart disease using patient data," Sci. Rep., 2025.

[16] S. Mirjalili and A. Lewis, "The whale optimization algorithm," Adv. Eng. Softw., vol. 95, pp. 51-67, 2016.

[17] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," Adv. Eng. Softw., vol. 69, pp. 46-61, 2014.

[18] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in Proc. IEEE Int. Conf. Neural Networks, vol. 4, 1995, pp. 1942-1948.

[19] H. M. Mohammed, S. U. Umar, and T. A. Rashid, "A systematic and meta-analysis survey of whale optimization algorithm," Comput. Intell. Neurosci., vol. 2019, p. 8718571, 2019.

[20] Q. Al-Tashi, S. J. Abdul Kadir, H. M. Rais, S. Mirjalili, and H. Alhussian, "Binary optimization using hybrid grey wolf optimization for feature selection," IEEE Access, vol. 7, pp. 39496-39508, 2019.

[21] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," Neurocomputing, vol. 172, pp. 371-381, 2016.
[22] Q. Li, H. Chen, H. Huang, X. Zhao, Z. Cai, C. Tong, W. Liu, and X. Tian, "An enhanced grey wolf optimization based feature selection wrapped kernel extreme learning machine for medical diagnosis," Comput. Math. Methods Med., vol. 2017, p. 9512741, 2017.
[23] M. H. Nadimi-Shahraki, H. Zamani, and S. Mirjalili, "Enhanced whale optimization algorithm for medical feature selection: A COVID-19 case study," Comput. Biol. Med., vol. 148, p. 105858, 2022.
[24] "A novel research on a heart disease prediction model based on hybrid intelligent optimization of deep networks," Discov. Networks, 2025.
[25] "Visual field prediction using a deep bidirectional gated recurrent unit network model," Sci. Rep., 2023.
[26] Z. C. Oleiwi, E. N. AlShemmary, and S. Al-Augby, "Developing hybrid CNN-GRU arrhythmia prediction models using Fast Fourier Transform on imbalanced ECG datasets," Math. Model. Eng. Probl., vol. 11, no. 2, pp. 413-429, 2024.
[27] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in Proc. 33rd Int. Conf. Machine Learning, PMLR 48, 2016, pp. 1050-1059.
[28] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," in Proc. 32nd Int. Conf. Machine Learning, PMLR 37, 2015, pp. 1613-1622.
[29] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?," Adv. Neural Inf. Process. Syst., vol. 30, pp. 5574-5584, 2017.
[30] "Uncertainty-aware diabetic retinopathy detection using deep learning enhanced by Bayesian approaches," Sci. Rep., 2025.
[31] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K. S. Kwak, "The internet of things for health care: A comprehensive survey," IEEE Access, vol. 3, pp. 678-708, 2015.
[32] "Recent advances on IoT-assisted wearable sensor systems for healthcare monitoring," Biosensors, vol. 11, no. 10, p. 372, 2021.