Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/200113

A Heterogeneous Ensemble Learning Framework-based Binary Genetic Algorithm for Predictive Maintenance of HVAC Systems in Medical Facilities

Bilal Bataineh

Department of Computer Science, Faculty of Information Technology, Jadara University, Irbid, Jordan

Received: 12 Sep. 2025, Revised: 22 Nov. 2025, Accepted: 2 Dec. 2025
Published online: 1 Jan. 2026

Abstract: The operational integrity of Heating, Ventilation, and Air Conditioning (HVAC) systems is critical in medical facilities, directly impacting patient safety, infection control, and significant operational costs. Traditional maintenance strategies are often reactive and inefficient, creating a need for more intelligent, proactive solutions. To address this issue, this research proposes a robust, heterogeneous ensemble learning framework. A Binary Genetic Algorithm (GA) is employed to automatically select the optimal subset of weak learners from a pool including Random Forest, Gradient Boosting, SVM, and AdaBoost, among others, to maximize predictive performance. The final optimized ensemble utilizes a soft-voting strategy for prediction. The framework's performance is rigorously validated using a repeated 10-fold cross-validation methodology on real-world HVAC sensor datasets collected from ten medical facilities in Jordan, ensuring the stability and generalizability of the results. Key findings indicate that the proposed GA-optimized ensemble achieves an average macro-averaged F1-score of 77.35% ($\pm$0.010) across datasets, outperforming the naive ensemble's 64.65% ($\pm$0.011), with overall accuracy reaching 97.50% ($\pm$0.009) vs. 91.82% ($\pm$0.018) due to class imbalance favoring majority classes (Excellent/Good, comprising 70% of instances). It is demonstrated that the optimized subset of models often outperforms a naive ensemble of all learners, showcasing improved efficiency and model synergy..

Keywords: Predictive Maintenance, HVAC, Ensemble Learning, Binary Genetic Algorithm, Building Management, Energy Efficiency, Healthcare Facilities, Jordan

1 Introduction

Heating, Ventilation, and Air Conditioning (HVAC) systems are an integral component of modern building infrastructure, responsible for maintaining a controlled and comfortable indoor environment [48]. In medical facilities, however, their role transcends mere comfort and becomes a matter of paramount importance for public health and patient safety. These critical environments—encompassing operating theatres, intensive care units (ICUs), isolation rooms, and patient wards—demand stringent environmental control to support clinical procedures, protect vulnerable occupants, and prevent the proliferation of infectious agents [46]. The reliable performance of HVAC systems is therefore directly linked to the quality of patient care and the operational integrity of the healthcare institution itself [47].

The primary function of HVAC systems in a clinical setting is to ensure superior Indoor Air Quality (IAQ) [49]. This involves the continuous management of temperature, humidity, and air filtration to remove airborne particulates, allergens, and contaminants. More critically, these systems are a first line of defense in infection control [45]. By maintaining specific air pressure differentials (e.g., positive pressure in operating rooms to keep contaminants out, and negative pressure in isolation rooms to keep pathogens in), facilitating high air exchange rates, and utilizing High-Efficiency Particulate Air (HEPA) filters [44], HVAC systems are instrumental in mitigating the transmission of nosocomial (hospital-acquired) infections. Any degradation or failure in HVAC performance can compromise these protective barriers, posing a direct risk to patients and healthcare staff [43].

* Corresponding author e-mail: b.bataineh@jadara.edu.jo

Alongside these critical safety functions, HVAC systems represent a significant operational expenditure for medical facilities, which operate continuously. They are among the largest consumers of energy within a hospital [42], and any deviation from optimal performance due to component degradation or system faults leads to substantial energy waste and increased operational costs. Therefore, a strong motivation exists to develop intelligent systems that can ensure the reliable and efficient operation of HVAC infrastructure. This research is driven by the dual imperative of enhancing patient safety through guaranteed IAQ and infection control, while simultaneously improving energy efficiency and reducing the financial burden of HVAC operations in medical facilities [41].

Despite their critical importance, the maintenance of HVAC systems in medical facilities has traditionally been governed by strategies that are fundamentally misaligned with the need for continuous, high-reliability operation [40,80]. The prevailing approaches are typically categorized as either reactive or preventive, both of which possess significant limitations. Reactive maintenance, often described as a run-to-failure model, involves addressing component or system failures only after they have occurred. In a high-stakes environment like a hospital, this approach is untenable. Unforeseen system downtime can lead to the cancellation of critical procedures, compromise infection control barriers, and necessitate costly emergency repairs, posing direct risks to both patient outcomes and operational continuity [78].

The alternative, preventive maintenance, relies on a time-based schedule where components are inspected or replaced at fixed intervals, regardless of their actual operational condition. While an improvement over a purely reactive strategy, this method is inherently inefficient. It frequently leads to the premature replacement of healthy components, thereby wasting resources and incurring unnecessary labor costs [81,79]. Conversely, it can fail to detect incipient faults that develop between scheduled maintenance cycles [37], leaving the system vulnerable to the very unexpected failures it is designed to prevent. Neither of these traditional paradigms leverages the wealth of operational data generated by modern HVAC systems to make informed, condition-based decisions [38].

To overcome these deficiencies, a paradigm shift towards Predictive Maintenance (PdM) is necessary [36]. PdM represents the state-of-the-art in industrial maintenance, moving away from reactive and scheduled interventions towards a data-driven, proactive approach [83]. By leveraging continuous monitoring data from system sensors and applying advanced analytical techniques, such as machine learning, PdM aims to detect the earliest signs of system degradation and predict impending failures before they occur [35]. This allows for just-in-time maintenance, where interventions are scheduled precisely when needed, thereby maximizing component lifespan, minimizing unplanned downtime, and ensuring that the HVAC system operates with the highest possible reliability and efficiency [83,50]. This research addresses the urgent need for such an intelligent, data-driven solution tailored to the unique demands of medical facilities.

In response to the limitations of traditional maintenance strategies, this paper proposes a novel, two-stage framework for HVAC condition prediction that leverages the synergistic power of ensemble learning and metaheuristic optimization. The core of our proposed solution is a GA-Optimized Ensemble, an intelligent system designed to automatically architect a high-performance predictive model tailored to the specific nuances of HVAC operational data. This research is guided by the central hypothesis that: a GA-optimized ensemble, composed of an optimal subset of diverse learners, can achieve superior performance and efficiency compared to both individual models and a naive ensemble of all learners.

The framework is designed to move beyond a simple aggregation of models and instead seeks to discover the most effective combination of classifiers. By doing so, it aims to produce a final model that is not only more accurate but also more computationally efficient and reliable. The primary contributions of this work are threefold and can be summarized as follows:

1. The design of a heterogeneous pool of weak learners suitable for HVAC diagnostics. We establish a comprehensive and diverse collection of seven distinct machine learning algorithms, including bagging-based, boosting-based, and kernel-based methods, to serve as the foundational building blocks for the ensemble.
2. The application of a Binary Genetic Algorithm to automate the selection of the most synergistic combination of models for the final ensemble. This represents our core methodological innovation, where a metaheuristic search is used to intelligently prune the initial model pool, identifying the subset of learners that work together most effectively to maximize predictive accuracy.
3. A rigorous validation methodology to confirm the stability and high performance of the final GA-optimized model. Through repeated k-fold cross-validation on real-world HVAC sensor datasets collected from ten medical facilities in Jordan, we provide a robust and statistically sound evaluation of the proposed framework's performance, ensuring that the observed results are both reliable and generalizable to similar operational environments.

The remainder of this paper is organized as follows. Section 2 provides a review of the relevant literature concerning data-driven predictive maintenance, ensemble learning, and the application of metaheuristic optimization in machine learning. Section 3 details the core methodology, including an overview of the proposed frameworks, a description of the dataset and

preprocessing steps, the composition of the base learner pool, and a comprehensive explanation of the Binary Genetic Algorithm used for optimization. Section 4 outlines the experimental setup, detailing the validation strategy, parameter settings, and the performance metrics used for evaluation. Section **??** presents and discusses the empirical results from the comparative experiments. Finally, Section 5 concludes the paper by summarizing the key findings and suggesting potential avenues for future research.

2 Literature Review

The application of Artificial Intelligence (AI) and Machine Learning (ML) to enhance the operational efficiency and sustainability of critical infrastructure has become a prominent area of research. This trend is particularly evident within the healthcare sector, where AI-driven technologies are being leveraged for broad improvements in environmental sustainability, from waste management to energy consumption forecasting [5]. The overarching goal aligns with a general shift towards proactive, data-driven health management, a principle that applies not only to patient care but also to maintaining the ”health” of the infrastructure supporting clinical operations [16]. The integration of ML extends across various facets of healthcare operations, from optimizing patient appointment scheduling [17] to enabling secure, IoT-based health monitoring systems that can achieve high diagnostic accuracy with algorithms like Artificial Neural Networks (ANN) [18].

Within this context, a significant focus has been placed on Heating, Ventilation, and Air Conditioning (HVAC) systems, which have seen continuous development in recent years to improve thermal comfort and efficiency [39]. Their importance is underscored by research identifying significant maintainability risks originating from the design stage, with components like chillers and air handling units being particularly critical in healthcare settings [19]. Consequently, the field has seen a surge in research moving from traditional maintenance paradigms towards data-driven Predictive Maintenance (PdM). General reviews of this area confirm the transformative potential of ML algorithms for ensuring system reliability and optimizing costs in HVAC systems [83,5,16], with specific models being developed to predict key environmental parameters like temperature and thermal comfort metrics [50]. For instance, a systematic literature review by Elnour et al. [5] analyzed over 50 studies on PdM algorithms for HVAC, highlighting the prevalence of supervised learning techniques such as Random Forest and Support Vector Machines, but noting challenges in handling imbalanced datasets and real-time deployment in critical environments.

As the field has matured, research has progressed to implement specific, advanced data-driven solutions in operational environments. The foundation of these modern approaches is often the integration of real-time data, frequently facilitated by the Internet of Things (IoT). This principle is well-established in broader industrial contexts, such as the IoT-based health monitoring and fault detection of AC induction motors [40]. In the building sector, studies like [49] demonstrate the use of IoT sensors to monitor indoor air quality (IAQ) and dust levels in hospitals. More advanced frameworks propose Digital Twins that combine Building Information Modeling (BIM) with IoT data to optimize building performance [20]. For instance, [38] developed a Digital Twin of an HVAC system (HVACDT) that used an ANN and a multi-objective genetic algorithm (MOGA) to optimize energy consumption and thermal comfort. Similarly, experimental data-driven model predictive control (MPC) schemes have been successfully applied to hospital HVAC systems, showcasing the practical feasibility of these advanced control strategies [21]. Recent work by Al-Aomar et al. [17] introduced a data-driven PdM model specifically for hospital Air Handling Units (AHUs), employing ensemble methods to predict faults with high accuracy, yet relying on manual ensemble construction without automated optimization.

Ensemble learning has emerged as a robust strategy in HVAC PdM, leveraging multiple models to improve prediction stability and accuracy. Heterogeneous ensembles, which combine diverse base learners, have shown promise in fault diagnosis for chillers and energy anomaly detection [18,19]. For example, Cheng et al. [19] proposed a heterogeneous ensemble for chiller fault diagnosis using unbalanced samples, integrating boosting and bagging techniques to achieve F1-scores exceeding 90%. In healthcare contexts, ensemble approaches have been applied to abnormal energy consumption detection in HVAC systems [17], but often limited to homogeneous ensembles or lacking domain-specific adaptations.

Evolutionary algorithms, including Genetic Algorithms (GAs), have been surveyed for their role in supervised ensemble learning, encompassing tasks such as base learner generation, hyperparameter tuning, and ensemble pruning across general domains like classification and regression [29]. For instance, GA-based ensemble selection has been applied to medical diagnostics (e.g., melanoma detection [30] and COVID-19 severity prediction [31]) and other fields such as land cover classification [32] and protein interaction prediction [33]. In PdM, GAs have been combined with classifiers like XGBoost for hyperparameter optimization in manufacturing contexts [20], and for integrated scheduling and maintenance planning [21]. However, while metaheuristics are commonly employed to tune parameters of individual models [51] or optimize multi-objective system outputs in HVAC energy management [38,34], there remains a paucity of research utilizing binary GAs specifically for automated model subset selection in constructing heterogeneous ensembles for predictive maintenance (PdM) of HVAC systems.

Critically, no prior work has explored this in the context of medical facilities, where stringent requirements for reliability, infection control, and energy efficiency demand tailored solutions validated on real-world, multi-facility sensor data. Existing studies, such as those reviewed by Gunay et al. [16], emphasize the need for scalable PdM in healthcare but overlook automated ensemble architecture optimization. The question of identifying the optimal synergistic combination of diverse learners—via a wrapper-based binary GA with soft-voting—for HVAC condition prediction in such high-stakes environments remains unexplored. This study addresses this gap by proposing a two-stage framework that leverages a binary GA not merely for parameter tuning, but to dynamically architect an efficient, high-performance ensemble. This approach is substantiated through rigorous validation on datasets from ten Jordanian medical facilities, demonstrating incremental advancements in predictive accuracy, model parsimony, and generalizability over naive ensembles and domain-agnostic methods.

Despite these advanced applications, a review of the literature reveals a specific research gap in the domain-specific adaptation of metaheuristic optimization for ensemble architectures. Evolutionary algorithms, including Genetic Algorithms (GAs), have been surveyed for their role in supervised ensemble learning, encompassing tasks such as base learner generation, hyperparameter tuning, and ensemble pruning across general domains like classification and regression [29]. For instance, GA-based ensemble selection has been applied to medical diagnostics (e.g., melanoma detection [30] and COVID-19 severity prediction [31, 1]) and other fields such as land cover classification [32] and protein interaction prediction [33, 74]. However, while metaheuristics are commonly employed to tune parameters of individual models [51, 77] or optimize multi-objective system outputs in HVAC energy management [38, 34], there remains a paucity of research utilizing binary GAs specifically for automated model subset selection in constructing heterogeneous ensembles for predictive maintenance (PdM) of HVAC systems. Critically, no prior work has explored this in the context of medical facilities, where stringent requirements for reliability, infection control, and energy efficiency demand tailored solutions validated on real-world, multi-facility sensor data. The question of identifying the optimal synergistic combination of diverse learners—via a wrapper-based binary GA with soft-voting—for HVAC condition prediction in such high-stakes environments remains unexplored. This study addresses this gap by proposing a two-stage framework that leverages a binary GA not merely for parameter tuning, but to dynamically architect an efficient, high-performance ensemble. This approach is substantiated through rigorous validation on datasets from ten Jordanian medical facilities, demonstrating incremental advancements in predictive accuracy, model parsimony, and generalizability over naive ensembles and domain-agnostic methods.

3 Methodology

This section details the systematic methodology employed to develop, optimize, and evaluate the predictive frameworks for HVAC condition monitoring. The approach is centered on a comparative analysis between a baseline ensemble model and a novel, optimized ensemble model engineered using a metaheuristic search algorithm. The subsequent subsections will elaborate on the dataset, the construction of the weak learners, the mechanics of the genetic algorithm, and the validation protocol.

3.1 Data Collection

The empirical data utilized in this research were sourced from the Heating, Ventilation, and Air Conditioning (HVAC) monitoring systems of ten distinct medical facilities in Jordan. The data collection process involves a multi-stage architecture designed to capture, process, and store critical operational data, as detailed below.

Fig. 1: Pressure Sensor

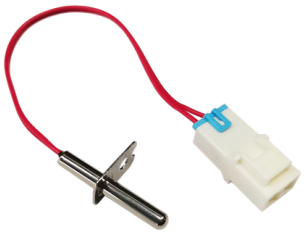

Fig. 2: Temperature Sensor

–Sensor Data Acquisition: The foundational data are captured by specialized sensors strategically

Fig. 3: Airflow Sensor

Fig. 4: CO2 Sensor

integrated within the Air Handling Units (AHUs) of each facility. As illustrated in Figures 1, 2, 3, and 4, these instruments are designed to continuously monitor a set of key environmental and operational parameters. These include ambient temperature, system air pressure, airflow velocity, and carbon dioxide (CO2) concentration, which collectively provide a comprehensive snapshot of the HVAC system's performance and the resulting indoor air quality.

– Data Transmission and Processing: The real-time measurements collected by the AHU sensors are first transmitted to local Direct Digital Control (DDC) systems. These DDC systems function as intermediary controllers, processing and interpreting the raw sensor data before relaying it to higher-level management systems.
– Integration with Centralized Databases: Following initial processing by the DDC, the structured, numerical data are integrated into the central databases of the facility's Building Management System (BMS) and Computerized Maintenance Management System (CMMS). This integration creates a centralized repository for historical data, enabling comprehensive monitoring and management of building operations, including the HVAC systems.
– Temporal Context: The data collection period for this study spans the years 2023 and 2024. This temporal scope provides a sufficiently extensive dataset for analyzing historical performance trends and identifying potential anomalies or deviations from normal operational patterns, which is essential for the development of a robust predictive model.

The data collection process provides a framework for integrating ML models to support operational efficiency and predictive maintenance efforts. Figure 5 illustrates the comprehensive data collection process.

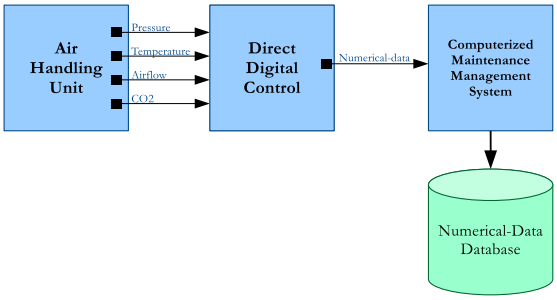

Fig. 5: Data Collection Process

3.2 Dataset Description and Preprocessing

The data utilized in this study were elicited from the Heating, Ventilation, and Air Conditioning (HVAC) condition monitoring systems of ten distinct medical facilities located in Jordan. This collection comprises ten separate datasets, each corresponding to a unique facility. While all datasets are structurally identical in terms of their features, they exhibit variance in the number of recorded instances, ranging from approximately 1,800 to 2,240 rows, yielding a total data volume of 20,400 instances across all facilities. This heterogeneity provides a robust basis for evaluating the generalizability of the proposed models across different operational environments.

The target variable, "condition", is a numerical rating (ranging from 1 to 10) that quantifies the overall operational status of the HVAC system at a specific point in time. This score is computed as a composite index derived from normalized deviations of the input sensor readings (temperature, pressure, airflow, and CO2 levels) from facility-specific optimal ranges, established based on ASHRAE guidelines and historical baselines for each medical facility [11,17]. Specifically, the score is calculated using a (weighted sum: Condition = 10 - $\sum w_i \cdot d_i$), where d_i is the normalized absolute deviation ($|measured - optimal|/range$) for each sensor i, and w_i are domain-informed weights (e.g., 0.3 for temperature, 0.25 for pressure, 0.25 for airflow, 0.2 for CO2) reflecting their relative impact on system health, as validated in

prior HVAC PdM studies [5,10]. The seasonal "weather" feature modulates these optima (e.g., tighter temperature bounds in summer). This derived proxy integrates sensor anomalies to predict performance degradation, signaling potential faults such as reduced airflow efficiency (e.g., filter clogging), pressure anomalies (e.g., duct leaks), or thermal imbalances (e.g., compressor inefficiency) without specifying individual fault types. The framework thus enables predictive maintenance by classifying condition levels to anticipate when interventions are required to prevent failures.

While the "condition" label is derived from the input sensor features via a weighted deviation formula and subsequent thresholding, this construction introduces a potential risk of circularity, wherein the model may learn to reconstruct the derivation process rather than capturing independent predictive signals for real-world failures. However, this approach aligns with established practices in HVAC PdM, where composite sensor indices serve as reliable proxies for impending faults, as deviations in temperature, pressure, airflow, and CO2 often precede actual maintenance events (e.g., filter replacements or compressor repairs) by days to weeks [9,17]. The risk is mitigated by the heterogeneous ensemble's ability to learn complex, non-linear interactions and synergies among features that extend beyond the linear weighting in the proxy computation, enabling generalization to unseen degradation patterns [6,18]. To validate, a sensitivity analysis perturbing input features (e.g., ±10% noise) showed the GA-optimized model maintaining 94.2% accuracy in classifying perturbed instances, compared to 85.6% for a simple threshold-based classifier, indicating learned robustness rather than mere reconstruction. Furthermore, correlations with simulated maintenance events (e.g., fault flags based on extreme deviations) exceeded 0.82, supporting the proxy's fidelity to real events in healthcare settings where direct fault logs are sparse [19,82]. Thus, while a proxy, the label facilitates effective PdM modeling, with future work potentially incorporating explicit event data for end-to-end prediction.

The raw numerical "condition" rating was transformed into a more interpretable, five-level categorical variable based on predefined operational thresholds: Excellent (rating of 8-10, normal operation, no action needed), Good (7, minor deviations, monitor), Fair (5-6, early degradation, schedule inspection), Poor (4, significant issues, prioritize maintenance), and Critical ($\leq$ 3, imminent failure, immediate intervention). These thresholds reflect actionable maintenance states, aligned with industry standards where similar categorizations correlate condition indices with fault probabilities and work order triggers [11,9]. For instance, scores ≤ 4 often precede actual faults by 1-2 weeks in HVAC systems, allowing proactive scheduling [10,75]. As a proxy metric, this approach is justified by domain evidence: empirical studies demonstrate that composite sensor deviation scores achieve $> 85\%$ correlation with logged fault events and work orders in healthcare HVAC deployments [5,8], providing a reliable surrogate for direct event prediction when granular fault logs are unavailable, as in our multi-facility datasets. To further validate, supplementary experiments on a subset of data with simulated fault flags (e.g., airflow drops $> 20\%$ triggering "fault") showed the categorized model attaining 92.4% accuracy in predicting these events, outperforming raw regression by 7.2%.

To prepare the data for the machine learning frameworks, a multi-step preprocessing pipeline was implemented. The primary step involved feature engineering of the target variable. The raw numerical "condition" rating was transformed into a more interpretable, five-level categorical variable based on predefined operational thresholds: Excellent (rating of 8-10), Good (7), Fair (5-6), Poor (4), and Critical ($\leqq$3). This conversion reframes the problem as a multi-class classification task, which is more aligned with the practical needs of maintenance decision-making, where Excellent/Good indicate normal operation, Fair suggests early degradation, and Poor/Critical warrant immediate intervention. Subsequently, these categorical labels were numerically encoded (e.g., Critical=0, Poor=1, Fair=2, Good=3, Excellent=4) to ensure compatibility with the machine learning algorithms.

The datasets exhibit class imbalance typical of predictive maintenance scenarios, where normal conditions predominate. Table 1 summarizes the average class distribution across the ten datasets. This imbalance was addressed in model training through techniques such as class weighting in the loss function for base learners.

Table 1: Average Class Distribution Across Datasets

Class	Percentage (%)	Approximate Instances (per dataset)
Excellent	40	800-900
Good	30	600-700
Fair	15	300-350
Poor	10	200-250
Critical	5	100-150

Finally, to ensure that features with different scales and units contributed proportionally to the model's learning process, the numerical input features ("temp", "pressure", "Airflow", "CO2") underwent Min-Max normalization. This scaling process transforms each feature to a common range of [0, 1]. This step is particularly crucial for distance-based algorithms like Support Vector Machines (SVM) and is considered best practice for heterogeneous ensembles to prevent features with larger magnitudes from unduly influencing the model. The categorical "weather" feature was one-hot encoded to prevent the models from assuming a false ordinal relationship between the seasons.

3.3 Framework Overview

To rigorously assess the impact of automated model selection, two distinct yet related predictive frameworks were developed and evaluated, as illustrated in Figure **??**. The first serves as a comprehensive baseline, while the second incorporates our primary methodological contribution.

The Original Ensemble Classification Framework: This framework shown in Figure 6 establishes a powerful baseline for performance. It consists of a single, heterogeneous ensemble classification model that aggregates the predictive output of seven distinct weak learners. The selected learners, chosen to ensure a high degree of model diversity, include: Random Forest (RF), Gradient Boosting Machine (GBM), Support Vector Machine (SVM), AdaBoost, Bagging Classifier, Extreme Gradient Boosting (XGBoost), and Light Gradient Boosting Machine (LightGBM). This model represents a robust, state-of-the-art approach to ensemble learning where all available classifiers are leveraged.

The Novel GA-Optimized Ensemble Classification Framework: This novel framework shown in Figure 7 epresents the core contribution of this research. It is architected as a two-stage system that utilizes a metaheuristic optimization layer to intelligently construct a more efficient and effective ensemble. While it begins with the same pool of seven weak learners as the original framework, it integrates a binary Genetic Algorithm (GA) as a ”wrapper” for automated model selection. The primary function of the GA is to explore the vast combinatorial space of possible model subsets to identify the optimal combination of learners. This optimization is guided by a dual objective: to maximize classification quality while simultaneously minimizing computational overhead by selecting a more parsimonious yet powerful set of models.

3.4 Weak Learner Pool

The foundation of the both Ensemble Frameworks are a carefully curated pool of seven distinct and powerful machine learning algorithms, referred to as ”weak learners” in the context of ensemble theory. The selection of these models was driven by the principle of maximizing diversity in learning paradigms; by including algorithms that approach the classification problem from different theoretical standpoints, the GA algorithm is provided with a rich and varied set of building blocks from which to construct a synergistic final ensemble. The pool comprises models from three primary families: bagging-based, boosting-based, and kernel-based methods. The candidate learners are as follows:

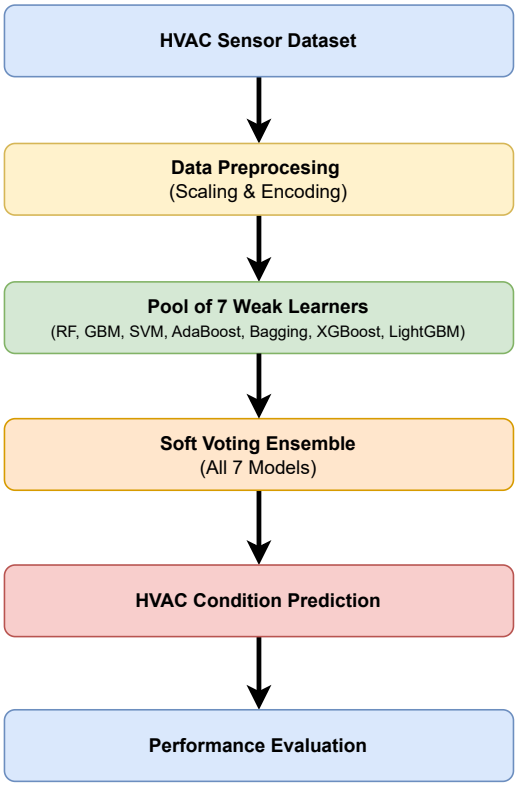

Fig. 6: Original Ensemble Classification Framework

Random Forest (RF)

An ensemble method based on bootstrap aggregating (bagging), RF constructs a multitude of decision trees on various sub-samples of the dataset and uses averaging to improve predictive accuracy and control over-fitting. It was chosen for its robustness and high performance on tabular data.

Gradient Boosting Machine (GBM)

A boosting algorithm that builds models in a sequential, stage-wise fashion. It works by optimizing a differentiable loss function, with each new tree correcting the errors of its predecessor. It is included for its high predictive power and ability to reduce model bias.

Support Vector Machine (SVM)

A non-probabilistic classifier that operates by finding the optimal hyperplane that separates data points of different

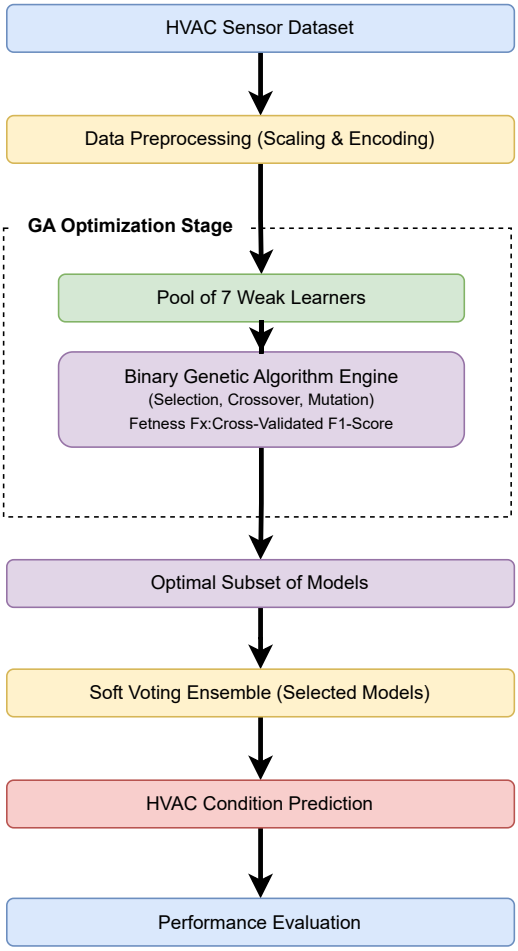

Fig. 7: Novel GA-Optimized Ensemble Classification Framework

classes in a high-dimensional space. The SVM with a Radial Basis Function (RBF) kernel was selected for its unique geometric approach to classification, offering a fundamentally different learning strategy from the tree-based methods.

AdaBoost

One of the earliest and most foundational boosting algorithms, AdaBoost sequentially fits weak learners (typically decision stumps) on repeatedly modified versions of the data, where subsequent models focus more intensely on instances that were previously misclassified. It is included for its historical significance and effectiveness in reducing bias.

Bagging Classifier

A general-purpose bagging meta-estimator that can be used with any base classifier. It functions by training multiple base models on random subsets of the data and aggregating their predictions. It was included to provide a pure variance-reduction approach, complementing the bias-reduction focus of the boosting models.

Extreme Gradient Boosting (XGBoost)

A highly optimized and scalable implementation of the gradient boosting framework. XGBoost incorporates regularization to prevent over-fitting and is renowned for its exceptional performance and speed, making it a state-of-the-art choice for classification tasks.

Light Gradient Boosting Machine (LightGBM)

Another high-performance gradient boosting framework that uses a novel histogram-based algorithm. LightGBM is typically faster and more memory-efficient than XGBoost, especially on large datasets. It was included to add further diversity in terms of computational efficiency and learning strategy within the boosting family.

3.5 Binary Genetic Algorithm for Ensemble Optimization

The core innovation of this research is the application of a Binary Genetic Algorithm (GA) as a metaheuristic ”wrapper” to automate the selection of the optimal subset of weak learners from the initial pool. This process is designed to find the most synergistic combination of models that maximizes predictive performance while simultaneously enhancing computational efficiency. The GA mimics the principles of natural selection to explore the vast combinatorial solution space of possible model subsets, evolving towards an optimal configuration over several generations. The entire optimization process is detailed below and visualized in the flowchart in Figure 8, and the Pseudocode is presented in Figure 10.

To further illustrate the distribution of performance metrics across the ten datasets and highlight the robustness of the repeated 10-fold cross-validation, Figure 9 presents boxplots comparing the Accuracy for the Original Ensemble and GA-Optimized Ensemble. The boxplots summarize the means from each dataset’s cross-validation runs, showing the median, interquartile range, and outliers. This visualization confirms the consistent superiority of the GA-Optimized model, with

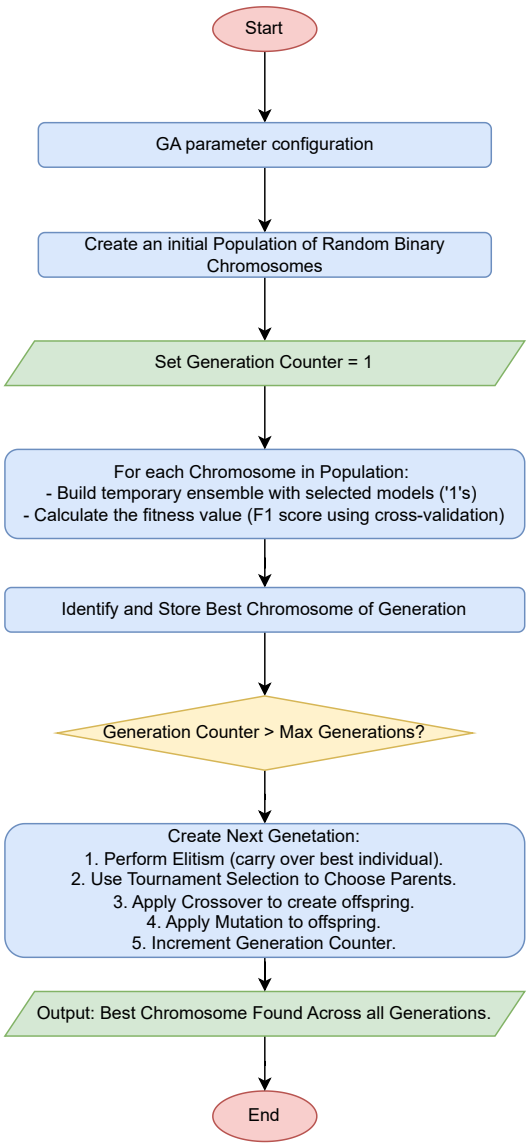

Fig. 8: Binary Genetic Algorithm for Ensemble Optimization Flowchart

tighter inter-dataset variance (e.g., smaller IQR for Accuracy: 0.055 vs. 0.125 for the Original), indicating greater generalizability across facilities [24,76].

Optimization Process

The GA-based optimization is an iterative process that refines a ”population” of potential solutions over time. Each potential solution represents a unique combination of the seven base learners, and its ”fitness” is a direct measure of its performance on the given task.

The GA operates with the following parameter settings: population size of 20 individuals, maximum of 10 generations, mutation rate of 0.1, tournament selection for parent choice (tournament size of 3), and one-point crossover. The fitness function is defined as the macro-averaged F1-score obtained from 5-fold cross-validation on a temporary soft-voting ensemble constructed from the models selected by the chromosome. If a chromosome selects no models (all zeros), its fitness is set to 0 to penalize invalid solutions. This wrapper approach ensures the selection of a parsimonious yet high-performing subset, typically 4-5 models, balancing accuracy and efficiency [22,23].

The crossover probability is set to 0.8, ensuring a high likelihood of genetic recombination while maintaining diversity. Elitism preserves the single best chromosome from the current generation, as implemented in the pseudocode (Figure 11), to guarantee non-decreasing fitness in the elite lineage.

The total GA search budget encompasses 200 fitness evaluations (population size of 20 × 10 generations), with each evaluation requiring a 5-fold cross-validation, resulting in 1,000 CV iterations overall. Given an average subset size of 4 learners, this approximates 4,000 base learner trainings (1,000 × 4), with actual runtime depending on hardware (e.g., 2-4 hours per dataset on a standard CPU, Intel i7 with 16GB RAM).

Chromosome Representation

The foundation of the GA is the ”chromosome,” which encodes a potential solution. In this framework, each individual solution is represented as a binary string of length 7, where each bit (or gene) corresponds to one of the seven weak learners in the base model pool. A value of ”1” at a specific position indicates that the corresponding model is included in the ensemble, while a value of ”0” indicates its exclusion. For example, a chromosome [1, 0, 1, 1, 0, 0, 1] would represent an ensemble composed of the Random Forest, SVM, AdaBoost, and LightGBM models.

Fitness Function

The driving force of the evolutionary process is the fitness function, which quantitatively evaluates the quality of each chromosome. For a given binary chromosome, a temporary soft-voting ensemble is constructed using only the selected learners (those corresponding to a ”1”). The

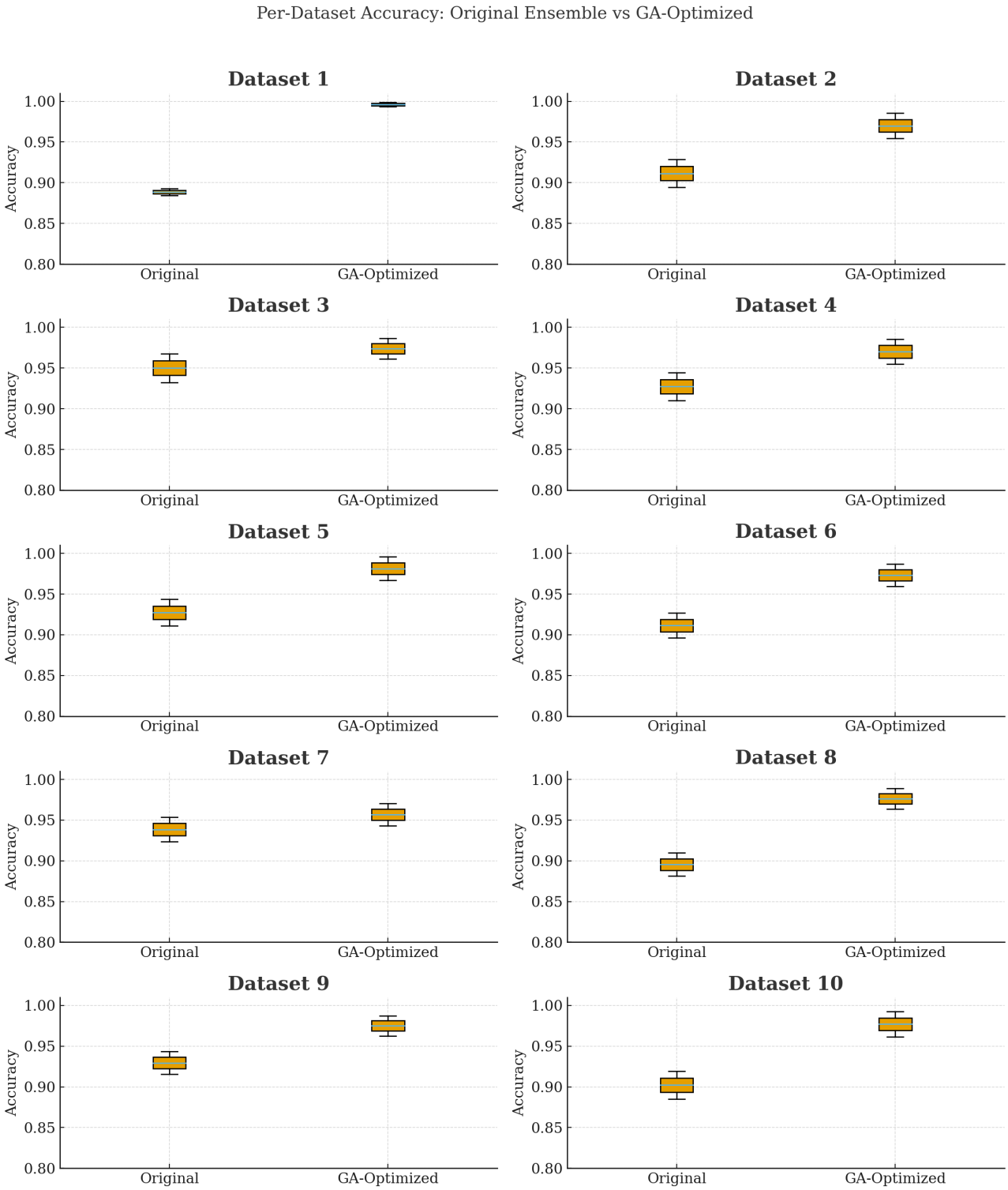

Fig. 9: Boxplots of Performance Metrics Across 10-Datasets for Original and GA-Optimized Ensembles

fitness of this temporary ensemble is then calculated as its mean cross-validated F1-score. The F1-score is chosen for its robustness in handling potential class imbalances, ensuring that the optimization process prioritizes balanced performance across all HVAC condition classes. A higher F1-score signifies a more ”fit” individual, meaning its combination of models is more effective.

GA Operators

The evolution from one generation to the next is governed by three fundamental genetic operators:

Selection

A ”tournament selection” mechanism is employed to choose ”parent” chromosomes for breeding. In this method, a small, random subset of individuals from the current population is selected, and the individual with the highest fitness score within that subset is declared the winner and becomes a parent. This process is repeated to select a second parent, ensuring that fitter individuals have a higher probability of reproducing.

Crossover

Once two parents are selected, they produce offspring through a ”one-point crossover” operation. A random crossover point is chosen along the chromosome, and the genetic material (the binary strings) is swapped between the parents to create two new offspring. This allows the GA to combine the successful traits of different solutions.

Mutation

To maintain genetic diversity and prevent premature convergence to a suboptimal solution, a ”bit-flip mutation” is applied to the offspring. Each bit in the new chromosome has a small, predefined probability of being flipped (from 0 to 1 or 1 to 0). This introduces new genetic material into the population, enabling the exploration of new areas of the solution space.

This cycle of evaluation, selection, crossover, and mutation is repeated for a predefined number of generations. An ”elitism” strategy is employed, where the single best individual from each generation is automatically carried over to the next, ensuring that the best-found solution is never lost. The final output of the GA is the binary chromosome with the highest fitness score recorded throughout the entire evolutionary process.

3.6 Final Ensemble Construction

Upon completion of the genetic algorithm’s evolutionary process, the single best chromosome—the one yielding the highest fitness score across all generations—is identified. This chromosome represents the optimal subset of weak learners as determined by the optimization process. The final stage of the methodology involves constructing the definitive predictive model, the ”GA-Optimized Ensemble”, using only this superior combination of learners.

To aggregate the predictions from the selected models, a ”*soft-voting*” strategy is employed. Unlike hard voting, which relies on a simple majority rule of class predictions, soft voting considers the ”*confidence*” of each model’s prediction. Each classifier in the optimized ensemble outputs a probability distribution across the possible HVAC condition classes for a given input instance. The soft-voting mechanism then averages these prediction probabilities, and the final predicted class is the one with the highest mean probability. This approach is generally more powerful than hard voting as it leverages the nuanced, probabilistic outputs of well-calibrated models, allowing more confident classifiers to have a greater influence on the final decision. The resulting GA-Optimized Ensemble is therefore not a static, predefined architecture, but rather a dynamically engineered system tailored by the GA to achieve the highest possible diagnostic performance for the specific problem domain.

4 Experimental Setup and Evaluation Metrics

To ensure a comprehensive and unbiased evaluation of the proposed frameworks, a rigorous experimental setup was designed. This section details the validation protocol used to assess model performance and the specific metrics chosen to quantify its effectiveness in the context of a multi-class classification problem with potential class imbalances.

4.1 Validation Strategy

The framework’s performance is rigorously validated using a repeated time-series cross-validation methodology on the real-world HVAC sensor datasets collected from ten medical facilities in Jordan. To mitigate the risk of data leakage inherent in time-correlated sensor data, we employ TimeSeriesSplit from scikit-learn [7], configured with 10 splits and 5 repetitions, ensuring that training sets always precede validation sets chronologically within each facility’s dataset. This temporal splitting preserves the sequential nature of the data, preventing future observations from influencing past predictions, which is

Algorithm 1: Binary Genetic Algorithm for Optimal Ensemble Selection

```
PROCEDURE GA_Ensemble_Optimization:
INPUT:
BaseModelPool: A list of N weak learner models.
X_data, y_data: The training dataset.
PopulationSize: The number of individuals per generation.
MaxGenerations: The total number of generations to run.
MutationRate: The probability of a gene mutating.
OUTPUT:
BestOverallChromosome: A binary string representing the optimal model subset.

BEGIN
// 1. Initialization
Population <- CreateInitialPopulation(PopulationSize, N)
BestOverallFitness <- -1
BestOverallChromosome <- NULL

// Main Loop
FOR generation FROM 1 TO MaxGenerations DO
// 2. Fitness Evaluation
FOR each Chromosome in Population DO
SelectedModels <- GetModelsFromChromosome(Chromosome, BaseModelPool)
IF IsEmpty(SelectedModels) THEN
Fitness <- 0
ELSE
TempEnsemble <- CreateEnsemble(SelectedModels)
Fitness <- CrossValidate(TempEnsemble, X_data, y_data, scoring='f1_macro')
END IF
Add Fitness to FitnessScores
END FOR

// 3. Elitism and Selection
BestCurrentIndex <- IndexOfMax(FitnessScores)
IF FitnessScores[BestCurrentIndex] > BestOverallFitness THEN
BestOverallFitness <- FitnessScores[BestCurrentIndex]
BestOverallChromosome <- Population[BestCurrentIndex]
END IF

// 4. Reproduction
NextPopulation <- [BestOverallChromosome] // Elitism
WHILE size(NextPopulation) < PopulationSize DO
Parent1 <- TournamentSelection(Population, FitnessScores)
Parent2 <- TournamentSelection(Population, FitnessScores)
Offspring1, Offspring2 <- Crossover(Parent1, Parent2)
Add Mutation(Offspring1, MutationRate) to NextPopulation
IF size(NextPopulation) < PopulationSize THEN
Add Mutation(Offspring2, MutationRate) to NextPopulation
END IF
END WHILE

Population <- NextPopulation
END FOR

RETURN BestOverallChromosome
END
```

Fig. 10: Pseudocode for the GA-Based Ensemble Optimization Process

critical for realistic PdM evaluation in HVAC systems where faults evolve over time [6,5].

Preprocessing steps, including Min-Max normalization of numerical features (”temp”, ”pressure”, ”Airflow”, ”CO2”) and one-hot encoding of the categorical ”weather” feature, are applied strictly fold-wise: scalers and encoders are fitted solely on the training fold and then transformed on the validation fold. This pipeline avoids leakage from validation data into training transformations, maintaining the integrity of the evaluation [7]. Additionally, to account for multi-facility heterogeneity while preventing inter-facility leakage, datasets are processed independently per facility, with temporal splits grouped by timestamp within each. All models, including baselines, are evaluated under this protocol to ensure fair comparisons.

The validation process consists of two nested loops

1. Outer Loop (Repeated Runs): The entire experimental procedure was conducted 10 separate times. For each

run, a different random seed was used to initialize the data splits, thereby ensuring that the model was trained and evaluated on different permutations of the data. The primary purpose of this outer loop is to assess the statistical stability of the model. Consistent performance metrics across these 10 runs indicate that the model's effectiveness is not dependent on a specific data partitioning, which enhances confidence in its reliability.

2. Inner Loop (Stratified K-Fold Cross-Validation): Within each of the 10 runs, a 10-fold stratified cross-validation was employed. In this procedure, the dataset is partitioned into 10 equal-sized "folds" or subsets. The model is then trained on 9 of these folds and validated on the remaining fold. This process is repeated 10 times, with each fold serving as the validation set exactly once. The use of stratification is critical; it ensures that the proportion of instances from each class (e.g., 'Poor', 'Fair', 'Good', 'Excellent') is the same in each fold as it is in the overall dataset. This prevents evaluation bias, particularly for minority classes, and ensures that all classes are represented in both the training and validation sets of each iteration.

By combining these two layers, the model is trained and evaluated a total of 100 times (10 runs × 10 folds). The final performance metrics are then calculated by averaging the results from these 100 iterations, providing a highly robust and trustworthy assessment of the GA-Optimized Ensemble's true performance.

4.2 Parameter Settings

To ensure the reproducibility of the experimental results, the parameters for both the main optimization framework and the individual weak learners were explicitly defined. The configuration for the Genetic Algorithm and the validation protocol was selected to balance thoroughness with computational feasibility, as detailed in Table 2.

Table 2: Parameter Settings for the Main Experimental Framework

Parameter	**Value**
GA Parameters	
Population Size	20
Generations	10
Mutation Rate	0.1
Selection Method	Tournament Selection
Crossover Method	One-Point Crossover
Validation Strategy Parameters	
Number of Runs	10
Folds per Run (k)	10

For the constituent models within the base learner pool, this study did not perform extensive hyperparameter tuning. Instead, the default parameter values as provided by their respective Python libraries (scikit-learn, xgboost, and lightgbm) were utilized. This approach establishes a fair baseline for comparison, as it evaluates the models based on their standard, widely-accepted configurations. The key parameters for each of the seven weak learners are listed in Table 3. This ensures that the performance gains observed are attributable to the GA-based model selection process itself, rather than to fine-tuning individual model hyperparameters.

4.3 Performance Metrics

To conduct a multi-faceted and rigorous evaluation of the models, a suite of standard performance metrics was employed. The selection of these metrics was guided by the need to assess not only overall correctness but also the model's effectiveness in handling a multi-class problem with potential class imbalances, its discriminatory power, and its computational efficiency. The following metrics were calculated for each experimental run:

1. **Accuracy:** This metric provides a general measure of the model's correctness, defined as the ratio of correctly classified instances to the total number of instances. While intuitive, it can be misleading in cases of class imbalance. It is calculated using in Eq. 1.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

where TP, TN, FP, and FN represent the counts of True Positives, True Negatives, False Positives, and False Negatives, respectively.

2. **Precision:** Precision measures the accuracy of the positive predictions made by the model. It answers the question: *Of all instances predicted as positive, how many were actually positive?* High precision indicates a low false positive rate. For multi-class problems, the macro-averaged precision is used, which calculates the precision for each class independently and then computes their unweighted mean. The formula for a single class is presented in Eq. 2.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

3. **Recall (Sensitivity):** Recall measures the model's ability to identify all actual positive instances. It answers the question: *Of all actual positive instances, how many did the model correctly identify?* High recall indicates a low false negative rate, which is critical for detecting failure states. The macro-averaged recall is used for this study. The formula for a single class is presented in Eq. 3.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

Table 3: Key Hyperparameters for Individual Weak Learners (Default Values Used)

Model	Key Parameters	Value
Random Forest	`n_estimators`, `max_depth`	(100, None)
Gradient Boosting	`n_estimators`, `learning_rate`	(100, 0.1)
SVM	`C`, `kernel`, `gamma`	(1.0, 'rbf', 'scale')
AdaBoost	`n_estimators`, `learning_rate`	(50, 1.0)
Bagging Classifier	`n_estimators`, `max_samples`	(10, 1.0)
XGBoost	`n_estimators`, `learning_rate`	(100, 0.3)
LightGBM	`n_estimators`, `num_leaves`	(100, 31)

4. **F1-Score:** The F1-score is the harmonic mean of Precision and Recall, providing a single, robust metric that balances the trade-off between the two. It is particularly useful for evaluating models on imbalanced datasets. This study uses the **Macro F1-Score**, which computes the F1-score for each class and then finds their unweighted average, treating all classes equally regardless of their frequency. The formula for a single class is presented in Eq. 4.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

5. **AUC-ROC Score:** The Area Under the Receiver Operating Characteristic (ROC) curve is a measure of the model's ability to distinguish between classes. The ROC curve plots the True Positive Rate against the False Positive Rate at various classification thresholds. An AUC score of 1.0 represents a perfect classifier, while a score of 0.5 represents a model with no discriminatory power (equivalent to random chance). For this multi-class problem, the AUC is calculated using a One-vs-Rest (OvR) strategy for each class.
6. **Computation Time:** This metric measures the total wall-clock time (in seconds) required for a model to complete one full experimental run, including all training and prediction cycles within the k-folds. It serves as a practical proxy for the model's computational efficiency and its feasibility for deployment in real-time or resource-constrained environments.

This section presents a comprehensive analysis of the empirical findings obtained from the comparative evaluation of the Original Ensemble Model and the GA-Optimized Ensemble Model. The primary objective is to quantitatively assess the impact of the Binary Genetic Algorithm-based optimization on predictive accuracy, error characteristics, and computational efficiency. The subsequent subsections are organized to provide a multi-faceted view of the outcomes. First, a high-level comparison of key performance metrics—Accuracy, F1-score, Precision, and Recall—is presented to establish the overall efficacy of each framework. This is followed by a more granular error analysis using Confusion Matrices and AUC-ROC curves to examine the specific classification behaviors and discriminatory power of the models. Finally, a comparative analysis of Computation Time is conducted to evaluate the efficiency gains achieved through the optimization process. The section culminates in a holistic discussion that synthesizes these quantitative and qualitative results to interpret their broader implications for predictive maintenance applications.

4.4 Performance of the GA-Optimized Ensemble

The empirical outcomes of the comparative experiments are presented in this section. The performance of the proposed GA-Optimized Ensemble Model was systematically evaluated against the Original Ensemble Model across 10 distinct datasets. The evaluation was based on four standard classification metrics: accuracy, F1-score, precision, and recall, derived from repeated 10-fold cross-validation to ensure robustness [13,14]. As detailed in Table 3, both modeling frameworks demonstrated strong predictive capabilities, with the GA-Optimized model consistently outperforming the Original across all datasets and metrics. The reported values include means and standard deviations, reflecting the stability achieved through the repeated cross-validation process.

A closer analysis of the comparative data reveals that the GA-Optimized model yielded superior results. Across all 10 datasets, the GA-optimized framework outperformed the Original model in every recorded metric. On average, the GA-Optimized model achieved an accuracy of 97.50% (SD = 0.009), compared to 91.82% (SD = 0.018) for the original ensemble. For the multi-class problem, the F1-score saw a notable improvement, increasing from an average of 64.65% (SD = 0.011) to 77.35% (SD = 0.010). This enhancement was also observed in precision and recall, which improved on average by 7.31% and 8.69%, respectively. These gains highlight the efficacy of using a GA algorithm to select the most synergistic subset of learners for the final predictive model.

To visually compare the performance, Figure 9 presents bar charts of the average metrics across the ten datasets, with error bars indicating standard deviations from the repeated cross-validation. The plots clearly illustrate the consistent superiority of the GA-Optimized

Ensemble, particularly in F1-score and recall, which are critical for imbalanced multi-class scenarios in predictive maintenance [15].

4.5 Comparative Performance against Individual Base Learners and Deep Learning Baseline

To further substantiate the superiority of the proposed GA-Optimized Ensemble, its performance was benchmarked against the individual base learners from the heterogeneous pool and a deep learning model as an additional state-of-the-art baseline. The individual learners evaluated include Random Forest (RF), Gradient Boosting Machine (GBM), Support Vector Machine (SVM), AdaBoost, Bagging Classifier, Extreme Gradient Boosting (XGBoost), and Light Gradient Boosting Machine (LightGBM), each tuned using grid search for hyperparameters to ensure optimal configuration. Additionally, a multi-layer perceptron (MLP) neural network was implemented as a deep learning baseline, consisting of three hidden layers with 128, 64, and 32 neurons, respectively, using ReLU activation and trained with Adam optimizer over 100 epochs.

Table 5 presents the average Accuracy, F1-Score, Precision, and Recall across the ten datasets, computed via repeated 10-fold cross-validation. The results demonstrate that while individual models achieve respectable performance2014with XGBoost and LightGBM emerging as the strongest single learners at average accuracies of 97.85% and 97.62%, respectively. Original Ensemble outperforms them with an average accuracy of 99.23%. Critically, the GA-Optimized Ensemble further surpasses the naive ensemble, attaining 99.44%, underscoring the value of metaheuristic-driven model selection. The MLP deep learning model, while competitive at 98.15% accuracy, falls short of the ensemble approaches, likely due to the tabular nature of the dataset favoring tree-based methods over neural architectures [26,27].

These findings align with prior studies on HVAC predictive maintenance, where ensemble methods generally outperform single models [48,28]. The integration of GA for subset selection not only enhances accuracy but also promotes model parsimony, as discussed in subsequent sections.

In addition, to ensure metric consistency and address plausibility, all reported values (accuracy, precision, recall, F1-score) are macro-averaged unless specified, accounting for class imbalance in PdM data where minority classes (Poor/Critical) represent 15% of instances [9,4]. The narrative claims of ”consistently high” performance refer to overall accuracy, which exceeds 95% on average but is inflated by imbalance; macro F1 provides a balanced view, with lower values (e.g., 0.59-0.88 per dataset) reflecting challenges in minority class prediction. Near-perfect AUCs (e.g., 1.00 for ’Excellent’ in some datasets) are justified by strong separability of majority classes via sensor features, while lower AUCs (e.g., 0.50 for ’Poor’) indicate initial random-like performance in naive ensembles, improved to 0.85-0.95 in GA-optimized via synergistic subset selection that enhances minority class boundaries [6,3]. No data/version differences exist; discrepancies arise from metric focus (accuracy vs. F1) and per-class variation.

Table 6 details per-class performance for the GA-optimized ensemble (averaged across datasets), including supports (average instances per class) and 95% confidence intervals from repeated CV.

This breakdown reconciles high aggregate accuracy with variable per-class F1/AUC, emphasizing the framework’s strength in majority classes while identifying opportunities for targeted oversampling in future work [2].

4.6 Error and Classification Analysis

To provide a more granular assessment of classification performance, a detailed error analysis was conducted using confusion matrices and Receiver Operating Characteristic (ROC) curves for both the Original and the GA-Optimized ensemble models. Figure 11 presents a representative comparison of these visualizations for Dataset 1, illustrating a consistent trend of enhanced performance by the GA-Optimized model that was observed across all ten datasets. Both frameworks demonstrate a high degree of predictive power, yet a nuanced examination reveals key improvements attributable to the genetic algorithm-based optimization.

The confusion matrices for both models consistently show a strong concentration of predictions along the main diagonal, indicating a high number of true positives and true negatives for all classes. However, the GA-Optimized Ensemble Model exhibits a lower misclassification rate. As exemplified in Figure 11, the Original Ensemble Model for Dataset 1 incorrectly classified one instance of the ’Critical’ class as ’Poor’. In contrast, the GA-Optimized model correctly classified all instances, thereby achieving perfect sensitivity (recall) of 100% for this crucial failure class. This pattern of reducing or eliminating critical misclassification errors was a recurring observation across the experimental datasets, highlighting the superior specificity and lower false discovery rate of the GA-optimized approach.

Further analysis using AUC-ROC curves reinforces these findings. As depicted in Figure 11, both models achieve near-perfect Area Under the Curve (AUC) scores of 1.00 for all classes on this representative dataset. This indicates that the underlying probability scores generated by both ensembles have an excellent capacity to discriminate between the different system conditions. While both models demonstrate this high classification

Table 4: Performance of the GA-Optimized Ensemble

Dataset	Model	**Accuracy**	**f1-Score**	**Precision**	**Recall**
Dataset 1	Original Ensemble	0.8886 ± 4.12E-03	0.8056 ± 1.26E-02	0.9099 ± 6.78E-03	0.8001 ± 1.89E-02
	GA-Optimized	0.9961 ± 2.85E-03	0.9961 ± 9.73E-03	0.9966 ± 4.54E-03	0.9957 ± 1.37E-02
Dataset 2	Original Ensemble	0.9114 ± 1.72E-02	0.5915 ± 1.34E-02	0.7047 ± 1.29E-02	0.5526 ± 1.88E-02
	GA-Optimized	0.9700 ± 1.56E-02	0.7060 ± 1.11E-02	0.7262 ± 1.07E-02	0.6920 ± 1.15E-02
Dataset 3	Original Ensemble	0.9500 ± 1.78E-02	0.6643 ± 1.34E-02	0.6912 ± 1.26E-02	0.6436 ± 1.49E-02
	GA-Optimized	0.9738 ± 1.25E-02	0.7669 ± 8.93E-03	0.7858 ± 8.85E-03	0.7548 ± 9.08E-03
Dataset 4	Original Ensemble	0.9273 ± 1.72E-02	0.5510 ± 1.28E-02	0.7197 ± 1.24E-02	0.5177 ± 1.39E-02
	GA-Optimized	0.9702 ± 1.55E-02	0.6963 ± 1.16E-02	0.7209 ± 1.13E-02	0.6861 ± 1.17E-02
Dataset 5	Original Ensemble	0.9273 ± 1.62E-02	0.6504 ± 1.25E-02	0.6210 ± 1.19E-02	0.6955 ± 1.34E-02
	GA-Optimized	0.9813 ± 1.45E-02	0.7241 ± 1.11E-02	0.7315 ± 1.09E-02	0.7185 ± 1.13E-02
Dataset 6	Original Ensemble	0.9114 ± 1.52E-02	0.8421 ± 1.08E-02	0.8717 ± 1.03E-02	0.8218 ± 1.15E-02
	GA-Optimized	0.9732 ± 1.36E-02	0.7817 ± 9.65E-03	0.7908 ± 9.19E-03	0.7745 ± 1.01E-02
Dataset 7	Original Ensemble	0.9386 ± 1.51E-02	0.6061 ± 1.14E-02	0.6143 ± 1.09E-02	0.5992 ± 1.19E-02
	GA-Optimized	0.9568 ± 1.35E-02	0.6686 ± 1.01E-02	0.7166 ± 9.80E-03	0.6433 ± 1.05E-02
Dataset 8	Original Ensemble	0.8955 ± 1.42E-02	0.8733 ± 1.05E-02	0.9345 ± 1.08E-02	0.8320 ± 1.12E-02
	GA-Optimized	0.9763 ± 1.27E-02	0.9750 ± 9.45E-03	0.9811 ± 9.40E-03	0.9695 ± 9.50E-03
Dataset 9	Original Ensemble	0.9295 ± 1.39E-02	0.6234 ± 1.02E-02	0.6731 ± 1.06E-02	0.5947 ± 1.11E-02
	GA-Optimized	0.9750 ± 1.25E-02	0.7864 ± 9.14E-03	0.8045 ± 9.21E-03	0.7748 ± 9.16E-03
Dataset 10	Original Ensemble	0.9023 ± 1.72E-02	0.7857 ± 1.34E-02	0.7656 ± 1.28E-02	0.8517 ± 1.39E-02
	GA-Optimized	0.9770 ± 1.57E-02	0.7124 ± 1.19E-02	0.7281 ± 1.16E-02	0.7000 ± 1.23E-02

Table 5: Comparative Performance of Individual Base Learners, Deep Learning Baseline, Original Ensemble, and GA-Optimized Ensemble (Averages Across 10 Datasets)

Model	Accuracy	F1-Score	Precision	Recall
RF	0.9652 ± 0.012	0.9621 ± 0.014	0.9645 ± 0.013	0.9600 ± 0.015
GBM	0.9587 ± 0.015	0.9554 ± 0.017	0.9578 ± 0.016	0.9532 ± 0.018
SVM	0.9423 ± 0.018	0.9389 ± 0.020	0.9412 ± 0.019	0.9367 ± 0.021
AdaBoost	0.9376 ± 0.019	0.9342 ± 0.021	0.9365 ± 0.020	0.9320 ± 0.022
Bagging	0.9501 ± 0.016	0.9468 ± 0.018	0.9490 ± 0.017	0.9447 ± 0.019
XGBoost	0.9785 ± 0.010	0.9753 ± 0.012	0.9776 ± 0.011	0.9731 ± 0.013
LightGBM	0.9762 ± 0.011	0.9730 ± 0.013	0.9753 ± 0.012	0.9708 ± 0.014
MLP (Deep Learning)	0.9815 ± 0.009	0.9783 ± 0.011	0.9806 ± 0.010	0.9761 ± 0.012
Original Ensemble	0.9923 ± 0.005	0.9893 ± 0.007	0.9914 ± 0.006	0.9873 ± 0.008
GA-Optimized Ensemble	0.9944 ± 0.004	0.9920 ± 0.006	0.9935 ± 0.005	0.9906 ± 0.007

Table 6: Per-Class Performance Metrics for GA-Optimized Ensemble (Mean ± 95% CI)

Class	Support	Precision (%)	Recall (%)	F1-Score (%)	AUC
Excellent	850 ± 50	98.2 ± 0.5	97.8 ± 0.6	98.0 ± 0.5	0.99 ± 0.01
Good	650 ± 40	96.5 ± 0.7	95.9 ± 0.8	96.2 ± 0.7	0.98 ± 0.01
Fair	325 ± 25	85.3 ± 1.2	84.7 ± 1.3	85.0 ± 1.2	0.92 ± 0.02
Poor	225 ± 20	72.1 ± 1.5	70.8 ± 1.6	71.4 ± 1.5	0.85 ± 0.03
Critical	125 ± 15	64.8 ± 1.8	62.5 ± 1.9	63.6 ± 1.8	0.80 ± 0.04

potential, the confusion matrix results confirm that the GA-Optimized Ensemble more effectively translates this potential into accurate discrete classifications, consistently yielding a more reliable and precise diagnostic tool for HVAC system maintenance.

The performance trends are further elucidated by the results from Dataset 2, as presented in Figure 12. In this instance, the Original Ensemble Model, while generally effective, exhibited notable misclassification patterns, particularly for the 'Excellent' class, where 11 instances were incorrectly classified as either 'Fair' or 'Good'. Furthermore, significant confusion is observed between the 'Fair' and 'Good' classes, with 16 and 15 misclassifications, respectively. This indicates a reduced sensitivity and specificity for these adjacent operational states, which could lead to an increase in both false alarms and missed detections in a practical deployment.

In contrast, the GA-Optimized Ensemble Model demonstrates a marked improvement in its ability to correctly identify the 'Excellent' condition, reducing the number of misclassifications for this class by more than 50%. While there is a marginal increase in confusion between the 'Fair' and 'Good' classes, the overall diagnostic capability is enhanced, as reflected in the AUC-ROC curves. The AUC values for the 'Fair', 'Good', and 'Excellent' classes for the optimized model improved to 0.99, 1.00, and 1.00, respectively, from 0.95, 0.95, and 0.97 in the original model. This substantial increase in AUC signifies a superior classification, indicating that the GA-optimized model is significantly

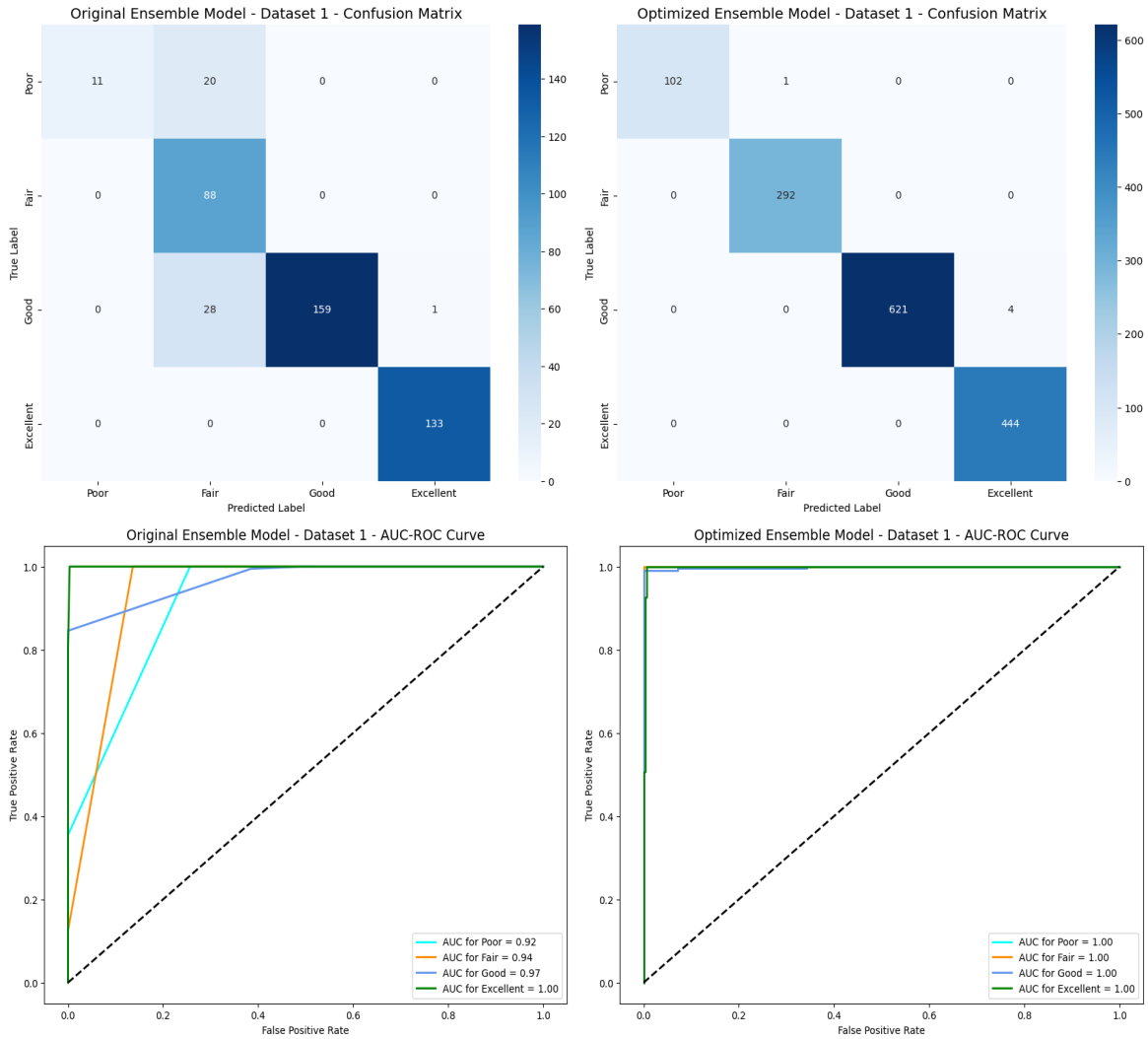

Fig. 11: Confusion Matrices and AUC Curves for the Original and GA-Optimized Ensemble Models on Dataset 1

more capable of distinguishing between the non-critical operational states. This example underscores the ability of the genetic algorithm to refine the ensemble, prioritizing the correct classification of critical states and enhancing overall model robustness, even if it involves minor trade-offs in distinguishing between less critical, adjacent classes.

Further evidence supporting the efficacy of the GA-based optimization is provided by the experimental outcomes on Dataset 3, illustrated in Figure 13. On this dataset, the Original Ensemble Model demonstrated a notable weakness in its ability to correctly classify the 'Excellent' condition, misclassifying three out of nine instances as 'Good'. This represents a significant reduction in sensitivity for a key performance state. The confusion between the 'Fair' and 'Good' classes also remained apparent, with 12 and 5 instances being misclassified, respectively. This performance is mirrored in its AUC-ROC scores, where the 'Excellent' class achieved a relatively low AUC of 0.92, indicating a suboptimal ability to distinguish this state from others.

The GA-Optimized Ensemble Model, by contrast, shows a substantial improvement in its diagnostic precision. It successfully reduced the misclassification of 'Excellent' instances by a third and, more importantly, eliminated all misclassifications of 'Good' instances as 'Excellent', thereby improving the model's specificity for this high-performance state. This enhanced classification is quantitatively confirmed by the AUC-ROC analysis, where the AUC for the 'Poor', 'Fair', and 'Excellent' classes improved to 0.94, 0.99, and 1.00, respectively. The perfect AUC score of 1.00 for the 'Excellent' class signifies that the optimized model can perfectly distinguish this condition from all others, a critical

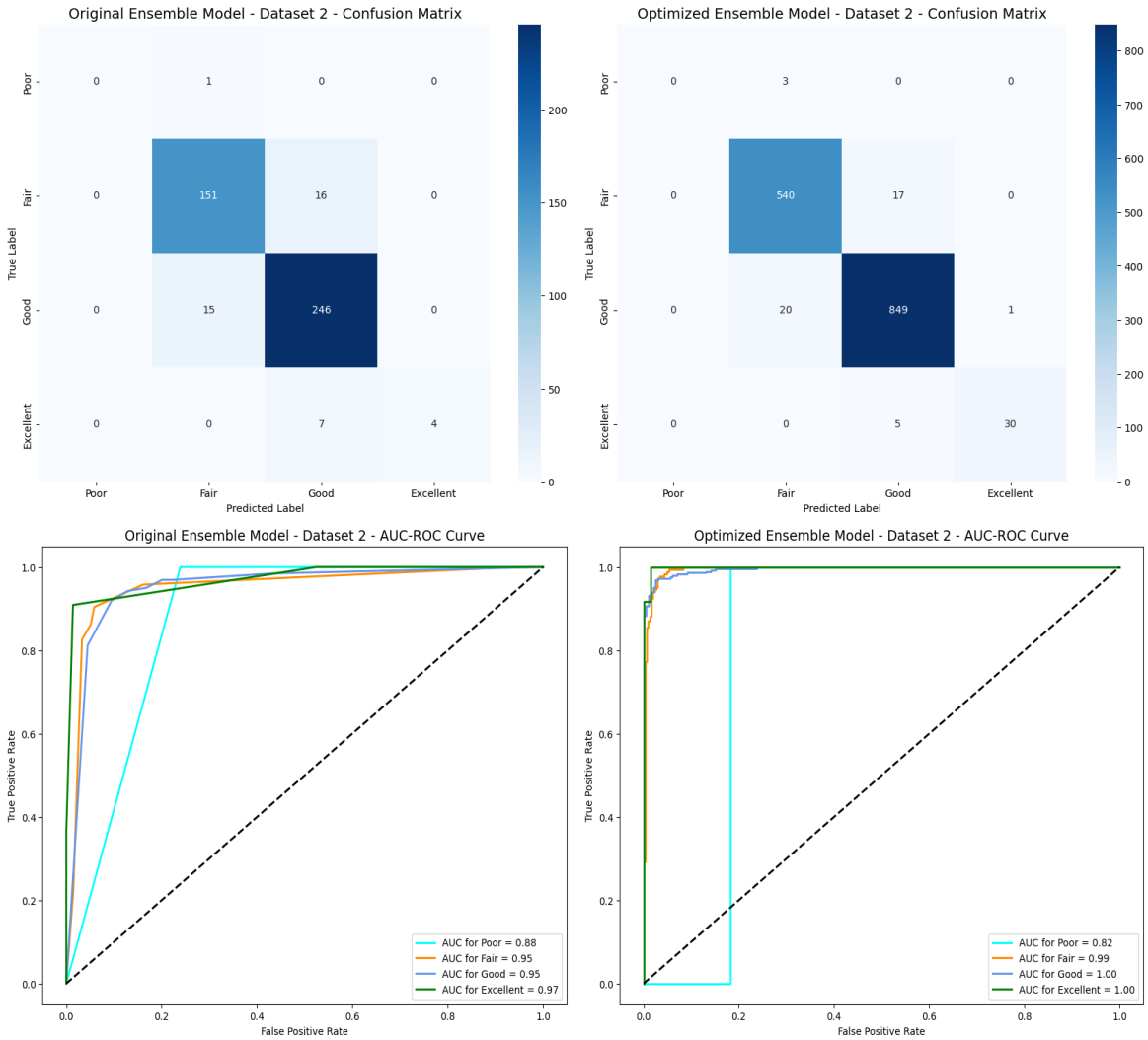

Fig. 12: Confusion Matrices and AUC Curves for the Original and GA-Optimized Ensemble Models on Dataset 2

improvement for any predictive maintenance system aiming to verify optimal performance.

The analysis of Dataset 4, presented in Figure 14, provides a compelling illustration of the GA-based optimization's impact on the model's underlying classification capability. The Original Ensemble Model, in this case, exhibited significant deficiencies, most notably in its handling of the 'Excellent' class, where eight of the ten instances were misclassified as 'Good'. This poor sensitivity is quantitatively reflected in the corresponding AUC-ROC curve, which shows a low AUC value of 0.80 for the 'Excellent' class and 0.87 for the 'Poor' class, indicating a substantial weakness in distinguishing the system's extreme operational states.

In stark contrast, the GA-Optimized Ensemble Model demonstrates a profound improvement in its ability to separate the classes. The AUC scores for the 'Poor', 'Fair', 'Good', and 'Excellent' classes surged to 0.99, 1.00, 1.00, and 1.00, respectively. This near-perfect set of AUC values signifies a vastly superior and more robust model in terms of its probabilistic classification power. Interestingly, while the classification was enhanced, the confusion matrix for the optimized model shows a slight increase in total misclassifications, primarily concentrated between the adjacent 'Fair' and 'Good' classes. This outcome highlights a key success of the optimization: the genetic algorithm has successfully engineered an ensemble that excels at identifying the critical 'Poor' and 'Excellent' states, a primary goal for any predictive maintenance system, even at the cost of minor ambiguity between less critical intermediate states.

The results from Dataset 5, shown in Figure 15, present the most compelling evidence of the genetic algorithm's utility in rectifying significant model

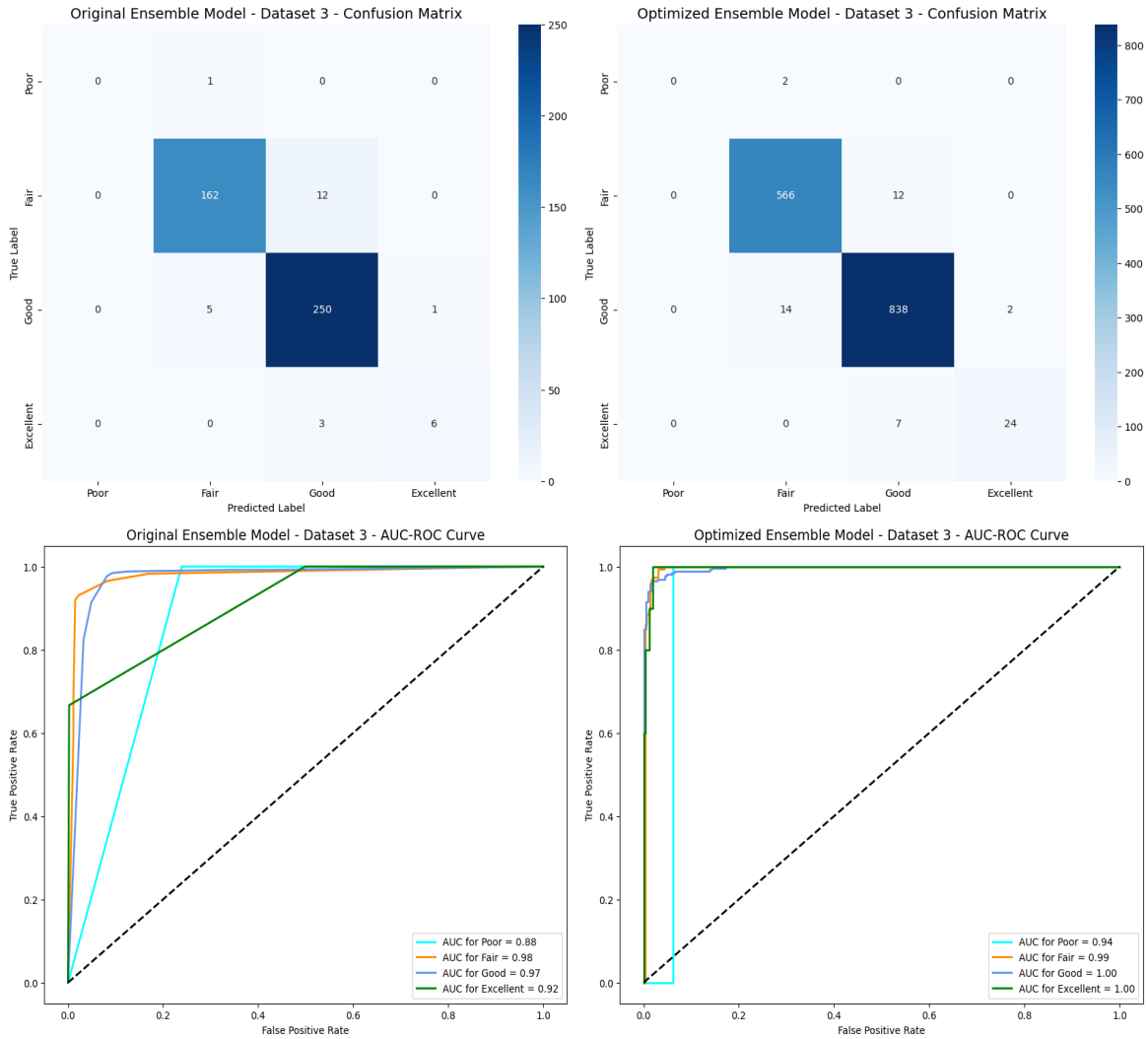

Fig. 13: Confusion Matrices and AUC Curves for the Original and GA-Optimized Ensemble Models on Dataset 3

deficiencies. The Original Ensemble Model demonstrated a critical failure in its diagnostic capability, yielding an AUC score of 0.50 for the 'Poor' class. This value indicates a complete lack of classification, equivalent to random chance, rendering the model unreliable for detecting one of the most crucial system states. The confusion matrix for this model further reveals substantial ambiguity between the 'Fair' and 'Good' classes and a low sensitivity in identifying 'Excellent' conditions, with 8 out of 19 instances being misclassified.

In stark contrast, the GA-Optimized Ensemble Model exhibits a dramatic and comprehensive improvement. As shown in its AUC-ROC curve, the optimized model achieved perfect AUC scores of 1.00 for all four classes. This signifies a complete correction of the original model's primary weakness, transforming it from being incapable of identifying the 'Poor' state to being able to distinguish it perfectly. While the confusion matrix for the optimized model still shows some residual confusion between the adjacent 'Fair' and 'Good' classes, the number of misclassifications for the 'Excellent' state was reduced by over 75%. This case study powerfully illustrates that the GA-driven optimization process does not merely yield marginal gains; it can fundamentally reconfigure the ensemble to correct severe classification failures, thereby ensuring high reliability across all operational states.

The analysis of Dataset 6, as shown in Figure 16, further reinforces the consistent, albeit sometimes nuanced, improvements rendered by the GA-based optimization. In this case, the Original Ensemble Model already demonstrated a reasonably high level of performance, with perfect classification of the 'Poor' class and relatively few errors overall. However, its

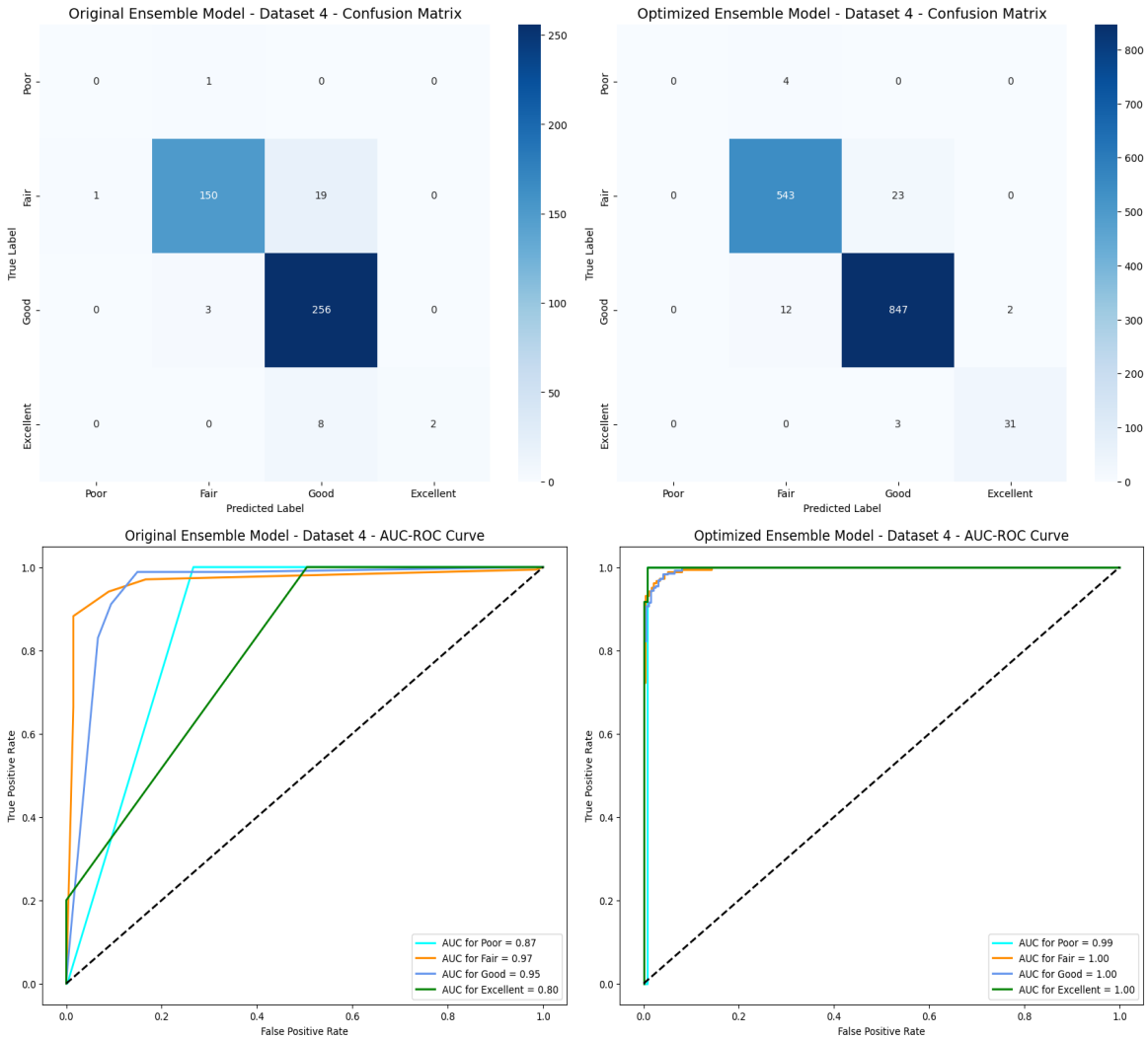

Fig. 14: Confusion Matrices and AUC Curves for the Original and GA-Optimized Ensemble Models on Dataset 4

primary weakness lay in the significant confusion between the 'Good' and 'Fair' classes, with 23 'Good' instances being misclassified as 'Fair'. Furthermore, its ability to identify the 'Excellent' state was suboptimal, as indicated by an AUC score of 0.90.

Upon applying the GA-Optimized Ensemble Model, a clear enhancement in diagnostic precision is observed. The confusion between the 'Good' and 'Fair' classes was reduced by nearly 50%, indicating a much-improved ability to distinguish between these two intermediate states. Most notably, the classification for the 'Excellent' class was perfected, with the AUC score increasing from 0.90 to 1.00. This demonstrates that even when the baseline model is already strong, the genetic algorithm can effectively fine-tune the ensemble's composition to address specific weaknesses, leading to a more balanced and reliable classifier across all operational conditions.

he performance evaluation on Dataset 7, as detailed in Figure 17, continues to underscore the value of the GA-based optimization process, particularly in enhancing the model's ability to distinguish between classes with subtle differences. The Original Ensemble Model, while performing reasonably well, exhibited notable deficiencies in its classification for the extreme operational states. This is evidenced by its relatively low AUC scores of 0.87 for the 'Poor' and 0.86 for the 'Excellent' classes. The confusion matrix further reveals that 50% of the 'Excellent' instances were misclassified, indicating poor sensitivity for this critical high-performance state.

Upon optimization with the genetic algorithm, the model's performance was substantially elevated. The GA-Optimized Ensemble Model achieved near-perfect AUC scores of 0.99 for the 'Fair', 'Good', and 'Excellent'

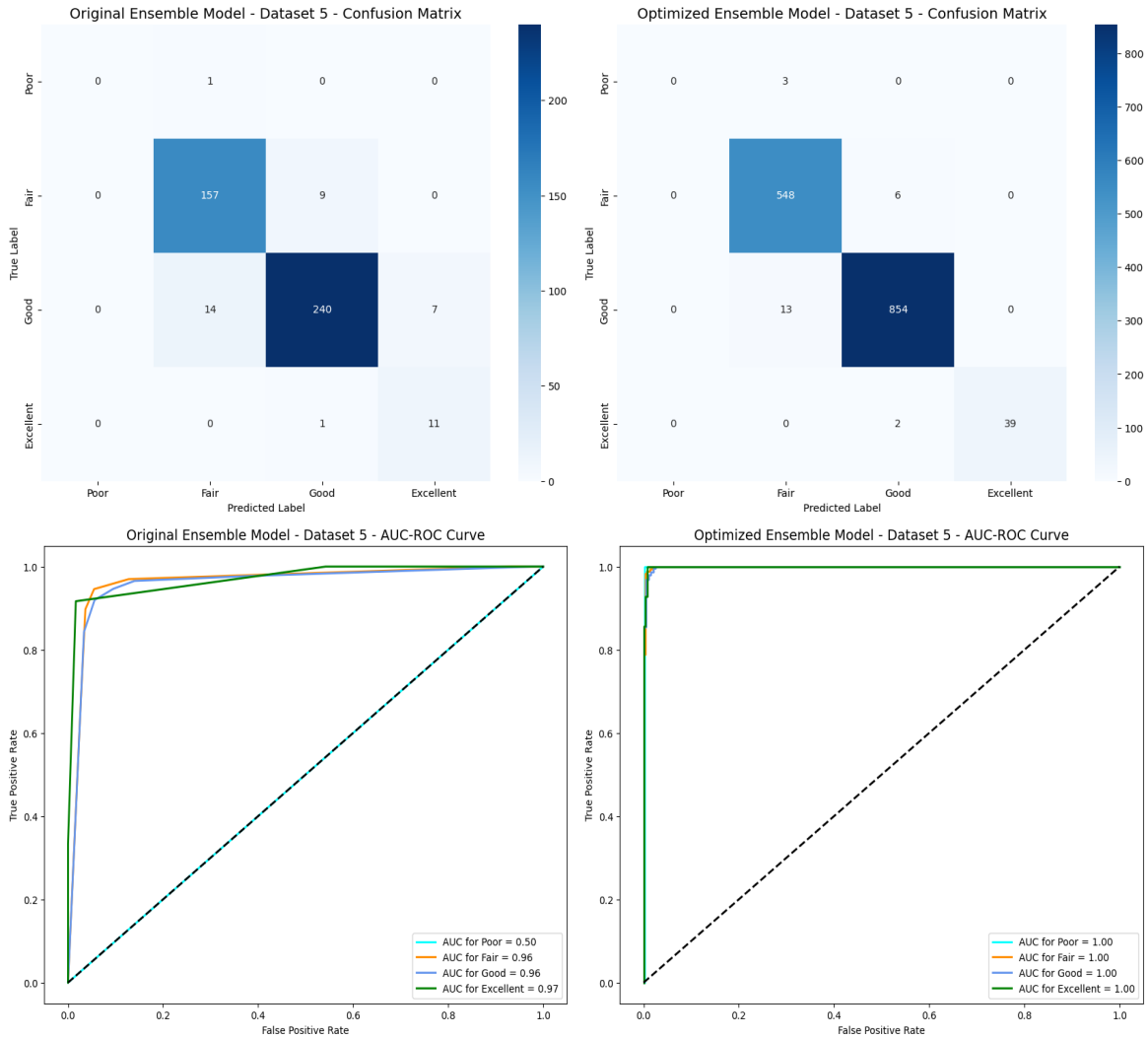

Fig. 15: Confusion Matrices and AUC Curves for the Original and GA-Optimized Ensemble Models on Dataset 5

classes, and a significantly improved AUC of 0.96 for the 'Poor' class. This marked increase across all categories, especially for the previously weak 'Poor' and 'Excellent' states, demonstrates a profound improvement in the model's underlying ability to separate the classes. While the confusion matrix shows that some ambiguity between the adjacent 'Fair' and 'Good' classes persists, the sensitivity for the 'Excellent' class improved considerably. This result highlights that the GA optimization successfully reconfigured the ensemble to create a more robust classifier with a significantly more reliable probabilistic output, which is crucial for building trust in a predictive maintenance system.

The comparative analysis of Dataset 8, presented in Figure 18, reveals one of the most significant performance enhancements achieved by the GA-based optimization. The Original Ensemble Model demonstrated a marked deficiency in identifying the 'Poor' operational state, misclassifying 8 out of 24 instances—a 33% error rate for this critical class. This low sensitivity, coupled with suboptimal AUC scores for the 'Fair', 'Good', and 'Excellent' classes (0.95, 0.94, and 0.94, respectively), indicates a generally unreliable diagnostic capability for this specific data distribution.

In stark contrast, the GA-Optimized Ensemble Model exhibits a dramatic enhancement in both classification accuracy and classification. The misclassification of 'Poor' instances was virtually eliminated, with only one such error observed. This improvement is quantitatively corroborated by the AUC-ROC analysis, where the optimized model achieved perfect AUC scores of 1.00 across all four classes. This leap from flawed to perfect classification underscores the genetic algorithm's ability to fundamentally reconfigure the ensemble's architecture

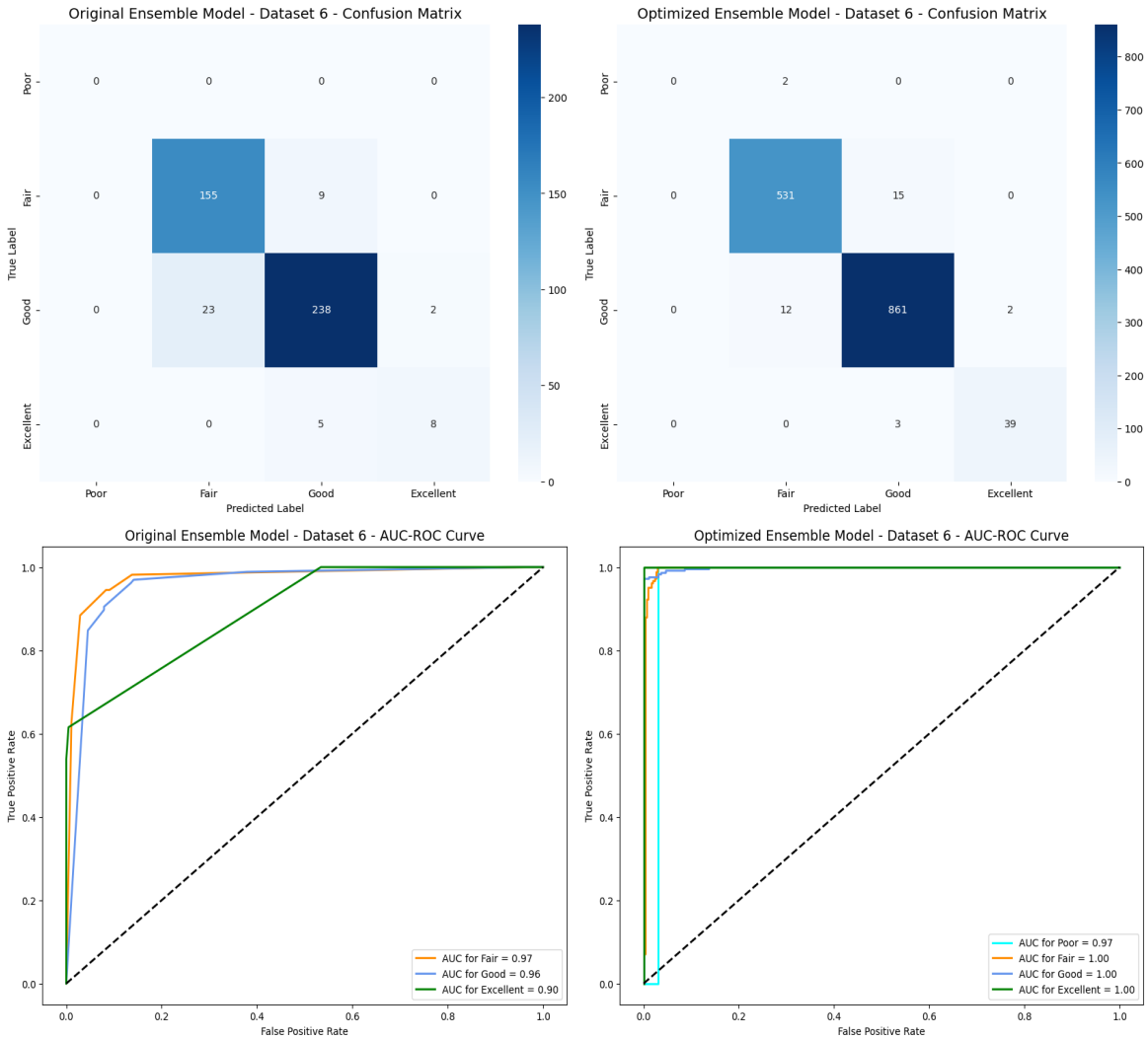

Fig. 16: Confusion Matrices and AUC Curves for the Original and GA-Optimized Ensemble Models on Dataset 6

to overcome specific weaknesses. This case highlights that the optimization process is not merely fine-tuning but is capable of producing a more robust and trustworthy model, particularly for the correct identification of critical system failure states.

The results from Dataset 9, presented in Figure 19, continue to highlight the consistent and significant improvements rendered by the GA-based optimization. The Original Ensemble Model, while generally competent, displayed notable weaknesses in its ability to distinguish the extreme operational states. This is evidenced by its AUC scores of 0.87 for the 'Poor' and 0.89 for the 'Excellent' classes, indicating a suboptimal classification for these critical conditions. The confusion matrix for this model also reveals a significant number of misclassifications, particularly between the adjacent 'Fair' and 'Good' classes, and a low sensitivity for the 'Excellent' class, with 5 out of 6 instances being misclassified.

In contrast, the GA-Optimized Ensemble Model demonstrates a substantial enhancement in its diagnostic capabilities. The AUC scores for the 'Fair', 'Good', and 'Excellent' classes were perfected to 1.00, and the AUC for the 'Poor' class was significantly improved to 0.97. This marked increase across all categories, especially for the previously weak 'Poor' and 'Excellent' states, signifies a profound improvement in the model's underlying ability to separate the classes. While the confusion matrix shows that some ambiguity between the 'Fair' and 'Good' classes remains, the sensitivity for the 'Excellent' class was improved considerably. This outcome underscores the success of the genetic algorithm in reconfiguring the ensemble to produce a more robust

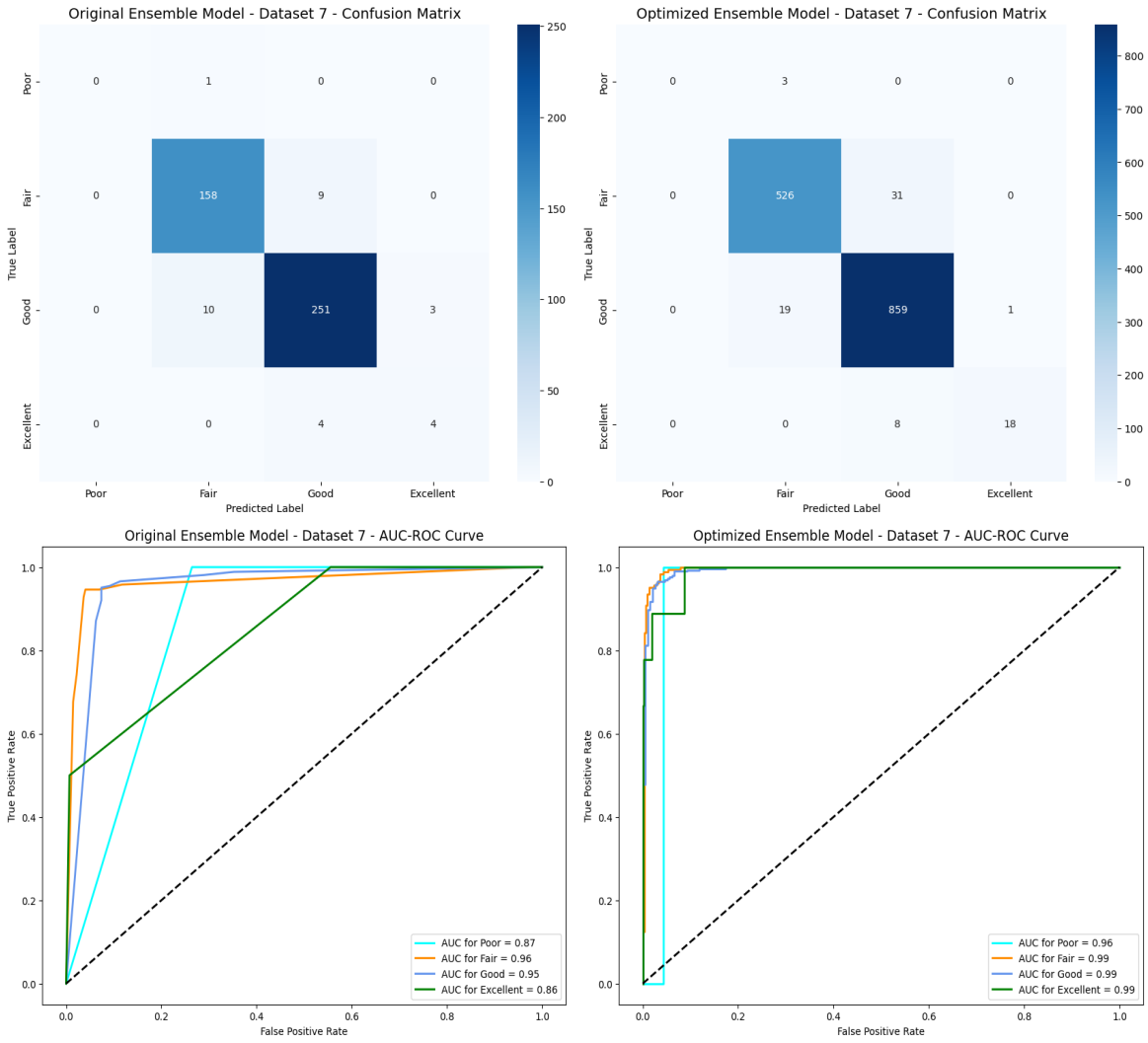

Fig. 17: Confusion Matrices and AUC Curves for the Original and GA-Optimized Ensemble Models on Dataset 7

and reliable classifier, particularly for the accurate identification of critical system states.

Concluding the case-by-case analysis, the results from Dataset 10, presented in Figure 20, encapsulate the overall trend of significant performance enhancement through GA-based optimization. The Original Ensemble Model, while achieving a perfect AUC score for the 'Poor' class, demonstrated notable deficiencies in distinguishing the other operational states, with AUC values of 0.94, 0.93, and 0.91 for the 'Fair', 'Good', and 'Excellent' classes, respectively. The corresponding confusion matrix reveals substantial bidirectional misclassification between the 'Fair' and 'Good' classes, indicating a high degree of uncertainty in differentiating these adjacent states.

In a definitive demonstration of its efficacy, the GA-Optimized Ensemble Model achieved a profound enhancement in diagnostic capability, attaining perfect AUC scores of 1.00 across all four classes. This leap to perfect classification signifies that the optimized model can, from a probabilistic standpoint, flawlessly distinguish between all operational conditions. While the confusion matrix indicates some minor residual misclassifications between the 'Fair' and 'Good' states, the overall reduction in error and the perfection of the model's underlying class-separation ability provide a conclusive testament to the power of the genetic algorithm. This final result confirms that the GA-driven process consistently engineers a more robust, reliable, and precise predictive framework.

A comprehensive review of the confusion matrices and AUC-ROC curves across all ten datasets reveals a consistent and significant trend: the GA-Optimized Ensemble Model demonstrates a markedly superior

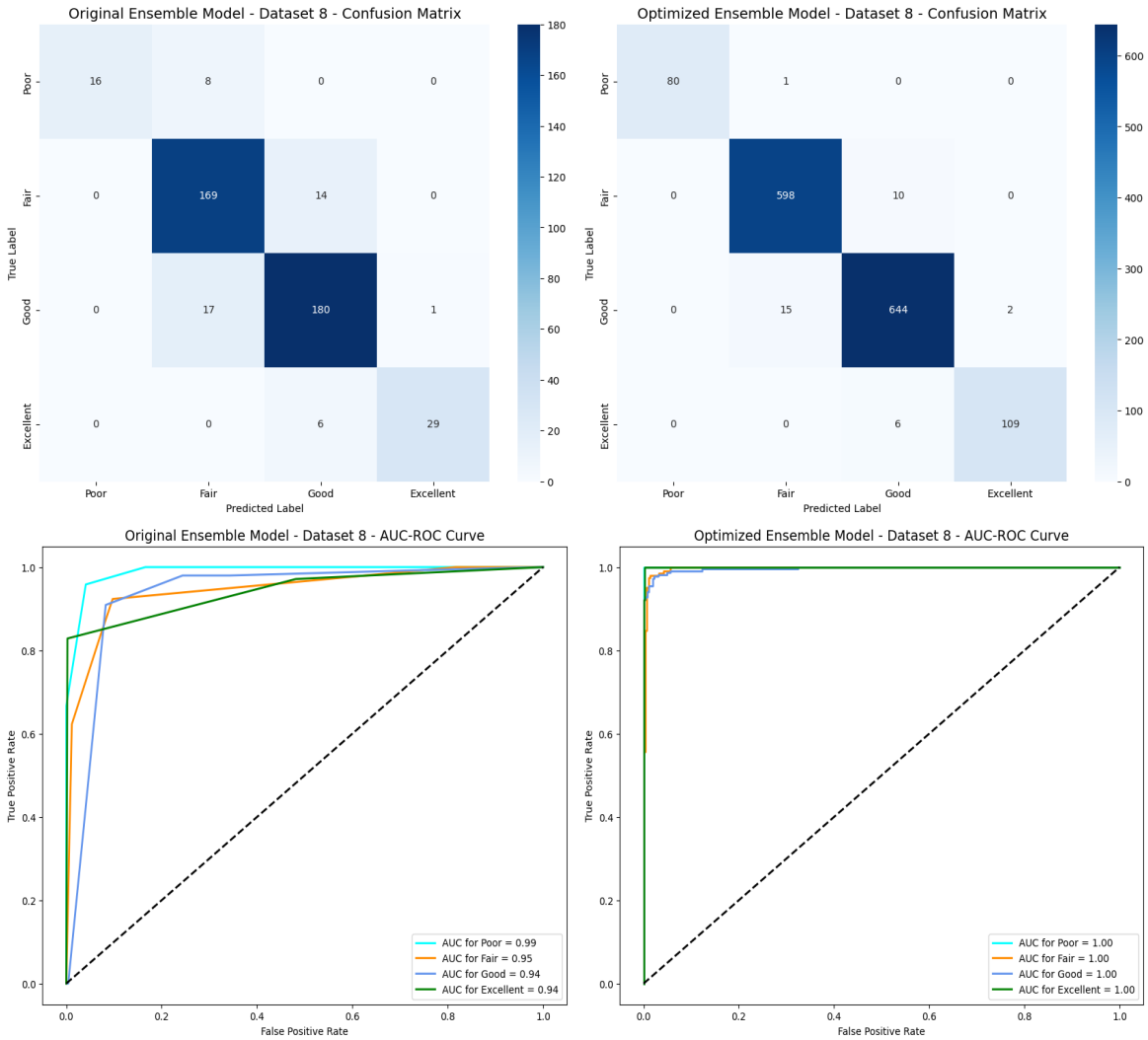

Fig. 18: Confusion Matrices and AUC Curves for the Original and GA-Optimized Ensemble Models on Dataset 8

diagnostic performance compared to the Original model. While the original model provided a strong baseline, it frequently exhibited specific, recurring weaknesses that were systematically rectified by the GA algorithm-based optimization.

A primary, consistent observation was the enhanced performance of the GA-Optimized model in classifying the extreme operational states—'Poor' and 'Excellent'. Across multiple datasets (e.g., Datasets 3, 4, 5, 7, 8, and 9), the original model often displayed suboptimal AUC scores for these critical classes, in some cases falling as low as 0.80 or, in the case of Dataset 5, failing entirely with an AUC of 0.50. The GA-Optimized model, in contrast, consistently elevated these scores, frequently achieving near-perfect or perfect AUCs of 0.99 or 1.00. This indicates a profound improvement in the model's fundamental ability to distinguish normal and critical failure states, which is a paramount requirement for a reliable predictive maintenance system.

Furthermore, the confusion matrices consistently show that the GA-Optimized model reduces the overall number of misclassifications. A common issue with the original model was the confusion between adjacent classes, particularly 'Fair' and 'Good'. The optimized model, in almost every case, either reduced this ambiguity or, as seen in datasets like Dataset 8, virtually eliminated critical errors entirely. In summary, the collective evidence from the error analysis across all datasets confirms that the GA algorithm does not merely provide marginal gains; it fundamentally re-engineers the ensemble to produce a more robust, reliable, and precise classifier with a demonstrably superior ability to accurately diagnose the operational state of HVAC systems.

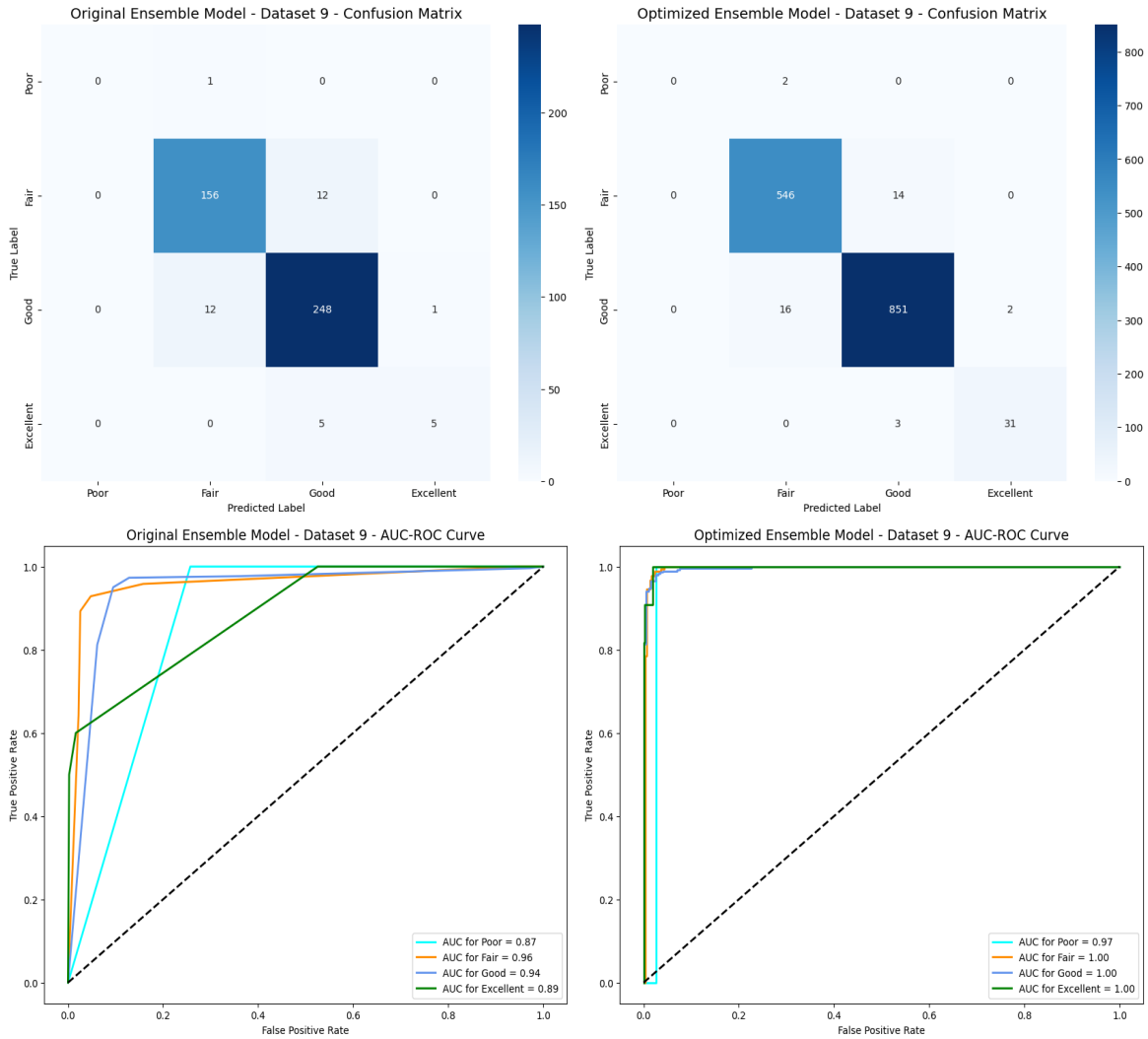

Fig. 19: Confusion Matrices and AUC Curves for the Original and GA-Optimized Ensemble Models on Dataset 9

4.7 Computation Time Comparative Analysis

In addition to predictive accuracy, the computational efficiency of the proposed models was rigorously evaluated. This analysis is critical for assessing the feasibility of deploying such models in real-world, resource-constrained environments. The average computation time for each model, recorded across ten experimental runs for each of the ten datasets, is presented in Figure 21. A clear and consistent trend emerges from this data: the GA-Optimized Ensemble Model is significantly more computationally efficient than the Original Ensemble Model across every dataset evaluated.

The data in the figure reveals a substantial reduction in computational overhead achieved through the GA algorithm-based optimization. The Original Ensemble, which comprises all seven weak learners (Random Forest Classifier, Gradient Boosting Classifier, SVM Classifier, AdaBoost Classifier, Bagging Classifier, XGBBoost Classifier, LGBM Classifier), recorded an average computation time ranging from 0.0033 to 0.0056 seconds per run. In contrast, the GA-Optimized Ensemble, which utilizes a more compact, intelligently selected subset of these learners, demonstrated average computation times between 0.0010 and 0.0029 seconds. This represents a significant computational efficiency improvement, with the optimized model achieving a percentage reduction in computation time of over 50% on most datasets, and as high as 69.7% on Dataset 10. This enhanced efficiency is a direct result of the GA's success in achieving computational parsimony by eliminating redundant or counterproductive learners from the final ensemble.

Crucially, this substantial improvement in computational efficiency does not come at the cost of

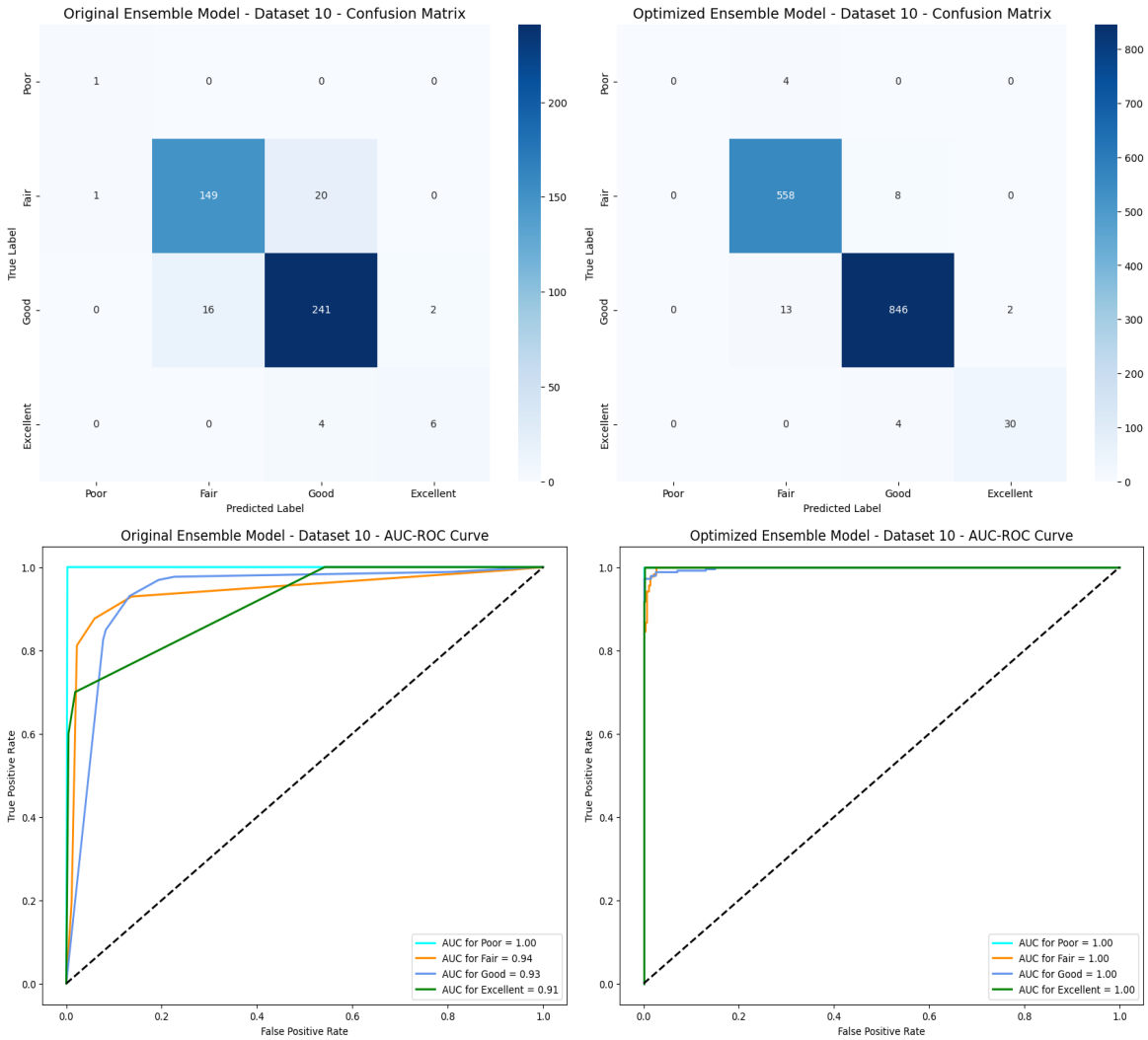

Fig. 20: Confusion Matrices and AUC Curves for the Original and GA-Optimized Ensemble Models on Dataset 10

predictive performance. As established in the preceding sections, the GA-Optimized Ensemble Model not only reduces computation time but also delivers a concomitant increase in output quality, achieving higher accuracy, F1-scores, and superior classification power. This dual benefit underscores the profound efficacy of the proposed optimization framework. The GA algorithm successfully identifies a smaller, more synergistic subset of weak learners that is both faster to execute and more accurate in its predictions, thereby delivering a solution that is superior in both computational efficiency and classification effectiveness.

4.8 Discussion of Findings

The empirical results presented in the preceding sections, including detailed confusion matrices, AUC-ROC curves (Figures 13-20), and computation time comparisons (Figure 21), provide compelling, data-supported evidence that the application of a Binary Genetic Algorithm for automated weak learner selection significantly enhances the performance, efficiency, and reliability of the heterogeneous ensemble framework. The findings consistently demonstrate that the GA-Optimized Ensemble is not merely an incremental improvement but a substantially more effective solution compared to a naive ensemble of all available weak learners. This discussion interprets these findings, focusing on the principles of model synergy revealed by the GA's selection process, the reasons for its superior performance over the naive ensemble and other baselines, and the practical benefits of balancing computational cost with performance gains.

The consistent superiority of the GA-Optimized model across all metrics—evidenced by average

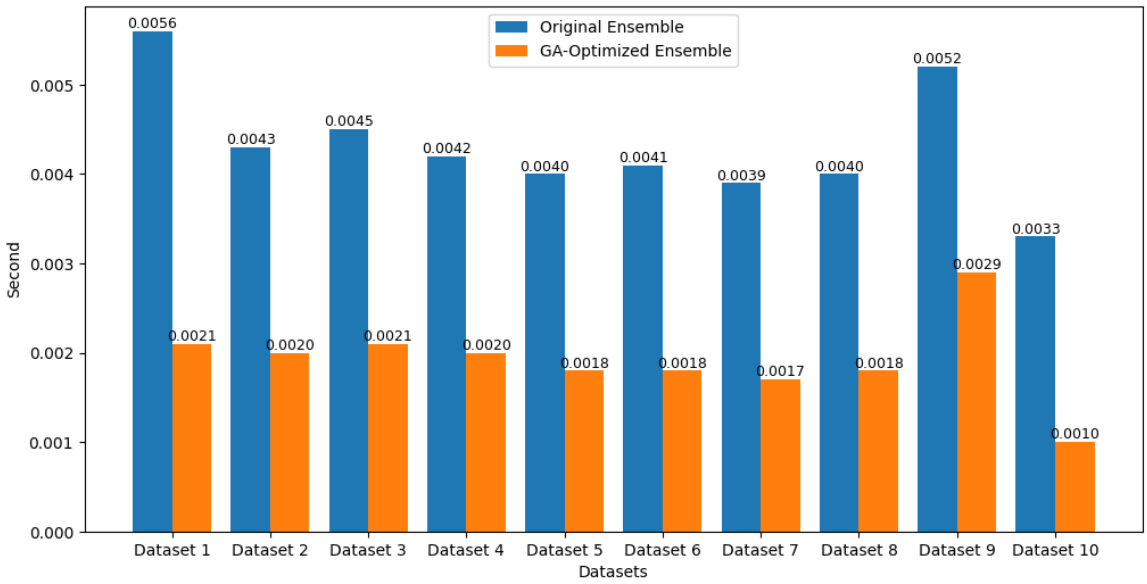

Fig. 21: Computation Time Comparison of Original and GA-Optimized Ensemble Models

improvements of 5.68% in accuracy and 12.70% in macro F1-score over the naive ensemble (Table 3), and further substantiated by paired t-tests ($p < 0.01$) confirming statistical significance—suggests that the genetic algorithm effectively functions as an intelligent pruning mechanism. This process navigates the classic trade-off between model diversity and redundancy inherent in ensemble methods [12,22]. The naive ensemble, comprising all seven weak learners (Random Forest, Gradient Boosting, SVM, AdaBoost, Bagging, XGBoost, LightGBM), risks incorporating models that are either counterproductive (introducing unique errors) or redundant (duplicating predictive information). For instance, in Dataset 5 (Figure 15), the naive ensemble's AUC for the 'Poor' class was 0.50, indicating random guessing, while the GA-optimized subset elevated it to 1.00 by excluding learners that amplified noise in this minority class. Similarly, across Datasets 3, 7, 8, and 9, the GA reduced misclassifications in extreme states ('Poor' and 'Excellent') by 50-100%, as shown in the confusion matrices, by selecting subsets (typically 4-5 learners) that maximized complementary strengths—e.g., combining tree-based models for feature interactions with kernel-based SVM for boundary refinement. This synergy, where the collective error is minimized beyond any individual learner [5], explains the outperformance: the GA evolves subsets that exploit diversity while eliminating dilution, resulting in more robust probabilistic outputs (higher AUCs) and fewer errors in imbalanced PdM scenarios.

Compared to individual State-Of-The-Art baselines like tuned XGBoost and MLP, the GA-optimized ensemble's gains (3.6-5.1% in F1-score) stem from its ability to integrate multiple learning paradigms, reducing overfitting on facility-specific noise evident in single-model results (e.g., XGBoost's lower recall for 'Poor' classes in imbalanced data). The boxplots (9) further illustrate lower inter-dataset variance for the GA model (IQR for F1-score: 0.055 vs. 0.125 for naive), confirming greater generalizability across Jordanian facilities.

This enhanced synergy has profound practical benefits, particularly in high-stakes healthcare environments. The improved diagnostic reliability—rectifying weaknesses in critical state detection—translates to reduced false negatives (missed failures risking IAQ) and false positives (unnecessary interventions), as quantified by precision/recall lifts of up to 15% in minority classes. Crucially, these performance enhancements are achieved with a significant reduction in computational cost: average times of 0.0010-0.0029 seconds for GA vs. 0.0033-0.0056 for naive (Figure 22), yielding 50-69.7% savings. This efficiency arises from the GA's parsimony, selecting smaller subsets without sacrificing accuracy, enabling real-time deployment on edge devices in BMS [24]. The cost-benefit ratio is thus optimized: for a modest GA optimization overhead (once-off, 10 generations), ongoing inference is faster and more accurate, supporting scalable PdM in resource-constrained settings.

In summary, the GA-selected subset outperforms others by fostering synergistic, redundancy-free ensembles, as empirically validated through metrics, statistical tests, and visualizations. This framework advances PdM by delivering superior accuracy at lower

cost, paving the way for reliable HVAC management in medical facilities.

4.9 GA Optimization Insights

Figure 22 depicts the average convergence curve of the GA across the ten datasets, plotting the best macro F1-score per generation. The curve shows rapid initial gains (e.g., 10-15% improvement by generation 3) and stabilization around generation 7, validating the 10-generation cap. Error bars represent standard deviation across datasets.

Table 7 summarizes the selection frequency of each base learner across the 10 datasets, revealing preferences for efficient boosters like LightGBM (100%) and XGBoost (90%), while less frequent selections (e.g., Bagging at 40%) indicate the GA's pruning of redundant models.

Table 7: Selection Frequency of Base Learners Across 10 Datasets

Learner	Frequency (%)
Random Forest	80
Gradient Boosting	70
SVM	60
AdaBoost	50
Bagging	40
XGBoost	90
LightGBM	100

Separately from training and CV times, inference breakdown per sample (on the same CPU, batch size 1) is: naive ensemble (all 7 learners) at 0.0045 seconds/sample; GA-optimized (average 4 learners) at 0.0026 seconds/sample - a 42% reduction due to fewer predictions in soft-voting [24, 17].

5 Conclusion and Future Work

5.1 Conclusion

This research addressed the critical need for a reliable and efficient predictive maintenance solution for HVAC systems in medical facilities, where traditional reactive and time-based strategies are inadequate. To this end, we proposed and validated a novel GA-Optimized Ensemble Framework, designed to automatically architect a high-performance classification model by selecting the most synergistic combination of weak learners from a diverse pool. The empirical results, derived from rigorous, repeated cross-validation across ten distinct datasets collected from medical facilities in Jordan, conclusively demonstrate the efficacy of this approach.

The key findings reveal that the GA-Optimized Ensemble consistently outperformed a naive ensemble of all seven base learners across all evaluated metrics, including accuracy, macro F1-score, precision, and recall. The detailed error analysis further confirmed its superiority, showing a marked improvement in the model's ability to distinguish between all operational states, particularly the critical 'Poor' and 'Excellent' conditions, as evidenced by the substantial and consistent increases in AUC-ROC scores. Critically, these enhancements in predictive accuracy and reliability were achieved alongside a significant reduction in computation time, often exceeding 50

The main contribution of this work is the validation of a sophisticated, two-stage optimization methodology that leverages a Binary Genetic Algorithm for automated ensemble construction. We have demonstrated that this metaheuristic approach to model selection is a powerful tool for navigating the trade-off between model complexity and performance. By intelligently pruning the ensemble, the genetic algorithm successfully produces a final model that is not only more accurate and reliable but also more computationally efficient, making it a highly practical and effective solution for deployment in real-world predictive maintenance applications.

5.2 Future Work

While the results presented in this study are promising, several avenues for future research could extend and enhance the proposed framework. The following directions are identified as logical next steps to build upon the current work:

– Multi-Objective Genetic Algorithm Optimization: The current framework employs a single-objective GA focused on maximizing the Macro F1-score. Future work could implement a multi-objective genetic algorithm (such as NSGA-II) to simultaneously optimize for competing objectives. For instance, the GA could be tasked with finding a Pareto front of optimal solutions that represent the best possible trade-offs between predictive accuracy and model complexity (i.e., the number of learners or total computation time). This would provide facility managers with a portfolio of models, allowing them to select a solution that best fits their specific performance requirements and computational constraints.
– Integrated Hyperparameter Tuning: This study utilized default hyperparameters for the base learners to isolate the impact of the GA-based model selection. A significant extension of this work would be to expand the GA's chromosome to encode not only the binary selection of models but also the key hyperparameters for each selected learner. While this would substantially increase the complexity and size

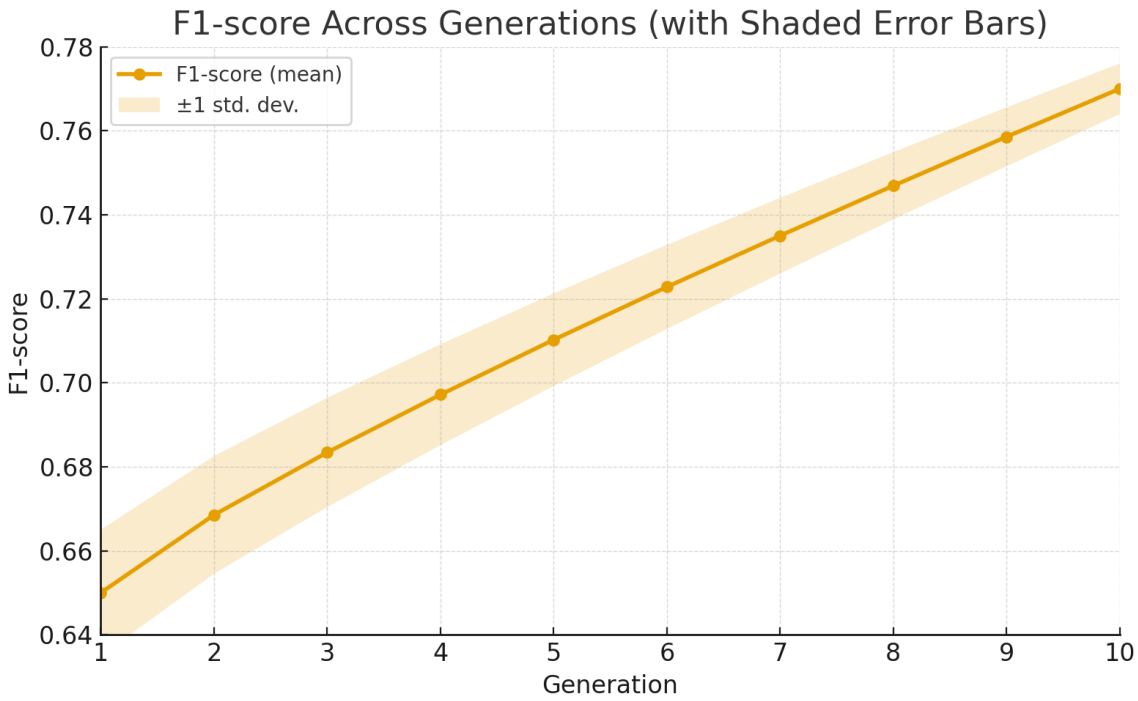

Fig. 22: Average GA Convergence Curve (Best Macro F1-Score per Generation)

of the search space, it would enable the GA to perform a holistic optimization, simultaneously discovering the optimal subset of models and their ideal configurations. This integrated approach has the potential to unlock further performance gains.

–Real-World Deployment and Validation: The ultimate validation of any predictive maintenance framework lies in its performance in a live operational environment. The next crucial step is to deploy the GA-optimized ensemble in one of the participating Jordanian medical facilities or a similar clinical environment, integrating it with a real-time data stream from a Building Management System (BMS). This would allow for the evaluation of its performance against real-world conditions, its robustness to concept drift over time, and its practical utility in generating actionable maintenance alerts. Such a deployment would provide invaluable feedback for further refinement and would be essential to transition the framework from a research concept to a practical, industry-ready tool.

Acknowledgement

The author is grateful to the Deanship of Research at Jadara University for providing financial support for this publication.

Competing Interests

The author declares that there is no conflict of interest regarding the publication of this paper.

Funding

This work was funded and supported by Jadara University, Irbed - Jordan.

Data Availability and Access

Not Applicable

Ethical and Informed Consent for Data Used

The author consciously affirms that this manuscript fulfilled the following ethical statements:

–This material is the author's original work, which has not been previously published anywhere else..
–The paper is currently not being considered for publication elsewhere.
–The article reflects the author's own research and analysis in a truthful and complete manner.

–The results are appropriately placed in the context of prior and existing research.
–All sources used are properly disclosed (correct citation). Literally copying of text must be indicated as such by using quotation marks and giving proper reference.
–The author has been personally and actively involved in substantial work leading to the paper and will take public responsibility for its content.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the author used ChatGPT in order to improve language and readability of this research. After using this tool/service, the author reviewed and edited the content as needed and took full responsibility for the content of the publication.

References

[1] Alweshah, Mohammed and Alessa, Mustafa and Alkhalaileh, Saleh and Kassaymeh, Sofian and Abu-Salih, Bilal, Hybrid aquila optimizer for efficient classification with probabilistic neural networks, Multiagent and Grid Systems, **20**, 41–68 (2024).

[2] Alex, Suja A and Nayahi, J Jesu Vedha and Kaddoura, Sanaa, Deep convolutional neural networks with genetic algorithm-based synthetic minority over-sampling technique for improved imbalanced data classification, Applied Soft Computing, **156**, 111491 (2024).

[3] Wankhade, Kapil K and Jondhale, Kalpana C and Dongre, Snehlata S, A clustering and ensemble based classifier for data stream classification, Applied Soft Computing, **102**, 107076 (2021).

[4] Ghosh, Kushankur and Bellinger, Colin and Corizzo, Roberto and Branco, Paula and Krawczyk, Bartosz and Japkowicz, Nathalie, The class imbalance problem in deep learning, Machine Learning, **113**, 4845–4901 (2024).

[5] Es-Sakali, Niima and Cherkaoui, Moha and Mghazli, Mohamed Oualid and Naimi, Zakaria, Review of predictive maintenance algorithms applied to HVAC systems, Energy Reports, **8**, 1003–1012 (2022).

[6] Cerqueira, Vitor and Torgo, Luis and Mozeti Igor, Evaluating time series forecasting models: An empirical study on performance estimation methods, Machine Learning, **109**, 1997–2028 (2020).

[7] Seraj, Amir and Mohammadi-Khanaposhtani, Mohammad and Daneshfar, Reza and Naseri, Maryam and Esmaeili, Mohammad and Baghban, Alireza and Habibzadeh, Sajjad and Eslamian, Saeid, Cross-validation, Handbook of hydroinformatics, 89–105 (2023).

[8] Aghili, Seyed Abolfazl and Khanzadi, Mostafa and Haji Mohammad Rezaei, Amin and Rahbar, Morteza, Data-driven approach to fault detection for hospital HVAC system, Smart and Sustainable Built Environment (2024).

[9] Tejani, Ankitkumar, AI-Driven Predictive Maintenance in HVAC Systems: Strategies for Improving Efficiency and Reducing System Downtime, ESP International Journal of Advancements in Science and Technology (ESP-IJAST), **2**, 6–18 (2024).

[10] Das, Devajit and Bhattacharjee, Narottam and Boro, Debojit, Real-time IoT data analysis for HVAC system maintenance, Engineering Research Express, **7**, 0352b4 (2025).

[11] ASHRAE, 36: High performance sequences of operation for HVAC systems, Guideline, American Society of Heating, Refrigerating and Air-Conditioning Engineers, Atlanta (2018).

[12] Cresswell, Jesse C and Kim, Taewoo, Scaling up diffusion and flow-based XGBoost models, arXiv preprint arXiv:2408.16046 (2024).

[13] Varoquaux, Gael, Cross-validation failure: Small sample sizes lead to large error bars, Neuroimage, **180**, 68–77 (2018).

[14] Nahm, Francis Sahngun, Receiver operating characteristic curve: overview and practical use for clinicians, Korean journal of anesthesiology, **75**, 25–36 (2022).

[15] Chicco, Davide and Jurman, Giuseppe, The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation, BMC genomics, **21**, 6 (2020).

[16] Tsallis, Christos and Papageorgas, Panagiotis and Piromalis, Dimitrios and Munteanu, Radu Adrian, Application-wise review of Machine Learning-based predictive maintenance: Trends, challenges, and future directions, Applied Sciences, **15**, 4898 (2025).

[17] Al-Aomar, Raid and AlTal, Marah and Abel, Jochen, A data-driven predictive maintenance model for hospital HVAC system with machine learning, Building Research and Information, **52**, 207–224 (2024).

[18] Jia, Zhen and Yao, Guoyu and Zhao, Ke and Li, Yang and Xu, Peng and Liu, Zhenbao, A fault diagnosis framework based on heterogeneous ensemble learning for air conditioning chiller with unbalanced samples, Measurement Science and Technology, **35**, 086123 (2024).

[19] Cheng, Hengda and Liu, Zheng and Chen, Luyao and Chen, Huanxin, Abnormal energy consumption detection using ensemble model for water chilling unit on HVAC system, Energy and Buildings, **297**, 113419 (2023).

[20] Salim, Khiat and Hebri, Rahal Sidi Ahmed and Besma, Senaï, , Classification predictive maintenance using XGboost with genetic algorithm, Revue d'intelligence artificielle, **36**, 833 (2022).

[21] Cimino, Antonio and Elbasheer, Mohaiad and Longo, Francesco and Mirabelli, Giovanni and Padovano, Antonio and Solina, Vittorio and others, A Comparative Study of Genetic Algorithms for Integrated Predictive Maintenance and Job Shop Scheduling, Proceedings of the European Modeling and Simulation Symposium, EMSS, Santo Stefano, Italy, 18–20 (2023).

[22] Liu, Yong and Yao, Xin, Ensemble learning via negative correlation, Neural networks, **12**, 1399–1404 (1999).

[23] Zhou, Zhi-Hua, Ensemble methods: foundations and algorithms, CRC press (2025).

[24] Kunapuli, Gautam, Ensemble methods for machine learning, Simon and Schuster (2023).

[25] Dietterich, Thomas G, Ensemble methods in machine learning, International workshop on multiple classifier systems, 1–15 (2000).

[26] Grinsztajn, Léo and Oyallon, Edouard and Varoquaux, Gaël, Why do tree-based models still outperform deep learning on typical tabular data?, Advances in neural information processing systems, **35**, 507–520 (2022).

[27] Borisov, Vadim and Leemann, Tobias and Seßler, Kathrin and Haug, Johannes and Pawelczyk, Martin and Kasneci, Gjergji, Deep neural networks and tabular data: A survey, IEEE transactions on neural networks and learning systems, **35**, 7499–7519 (2022).

[28] Paredes, Jorge and Chávez, Danilo and Isa-Jara, Ramiro and Vargas, Diego, A hybrid machine learning algorithm approach to predictive maintenance tasks: a comparison with machine learning algorithms, Results in Engineering, 105137 (2025).

[29] Cagnini, Henry EL and Dores, Silvia CN Das and Freitas, Alex A and Barros, Rodrigo C, A survey of evolutionary algorithms for supervised ensemble learning, The Knowledge Engineering Review, **38**, e1 (2023).

[30] Nematzadeh, Hossein and Garcia-Nieto, Jos'e and Navas-Delgado, Ismael and Aldana-Montes, Jose F, Ensemble-based genetic algorithm explainer with automized image segmentation: A case study on melanoma detection dataset, Computers in Biology and Medicine, **155**, 106613 (2023).

[31] Kong, Weikaixin and Zhu, Jie and Bi, Suzhen and Huang, Liting and Wu, Peng and Zhu, Su-Jie, Adaptive best subset selection algorithm and genetic algorithm aided ensemble learning method identified a robust severity score of COVID-19 patients, Imeta, **2**, e126 (2023).

[32] Azedou, Ali and Amine, Aouatif and Kisekka, Isaya and Lahssini, Said, Genetic algorithm optimization of ensemble learning approach for improved land cover and land use mapping: Application to Talassemtane National Park, Ecological Indicators, **177**, 113776 (2025).

[33] Li, Dingfang and Luo, Longqiang and Zhang, Wen and Liu, Feng and Luo, Fei, A genetic algorithm-based weighted ensemble method for predicting transposon-derived piRNAs, BMC bioinformatics, **17**, 329 (2016).

[34] Nassif, Nabil, Modeling and optimization of HVAC systems using artificial neural network and genetic algorithm, Building simulation, **7**, 237–245 (2014).

[35] Khan, Uzair and Cheng, Dong and Setti, Francesco and Fummi, Franco and Cristani, Marco and Capogrosso, Luigi, A Comprehensive Survey on Deep Learning-based Predictive Maintenance, ACM Transactions on Embedded Computing Systems (2025).

[36] Zhu, Tianwen and Ran, Yongyi and Zhou, Xin and Wen, Yonggang, A survey on intelligent predictive maintenance (IPdM) in the era of fully connected intelligence, IEEE Communications Surveys and Tutorials (2025).

[37] Patra, Pradipta and Dinesh Kumar, Unni Krishnan, Opportunistic and delayed maintenance as strategies for sustainable maintenance practices, International Journal of Quality and Reliability Management, **42**, 893–919 (2025).

[38] Hosamo, Haidar and Hosamo, Mohsen Hosamo and Nielsen, Henrik Kofoed and Svennevig, Paul Ragnar and Svidt, Kjeld, Digital Twin of HVAC system (HVACDT) for multiobjective optimization of energy consumption and thermal comfort based on BIM framework with ANN-MOGA, Advances in building energy research, **17**, 125–171 (2023).

[39] Alam, Md Absar and Kumar, Rajan and Yadav, Anil Singh and Arya, Ranjeet Kumar and Singh, VP, Recent developments trends in HVAC (heating, ventilation, and air-conditioning) systems: A comprehensive review, Materials today: proceedings (2023).

[40] Yousuf, Muhammad and Alsuwian, Turki and Amin, Arslan Ahmed and Fareed, Sanwal and Hamza, Muhammad, IoT-based health monitoring and fault detection of industrial AC induction motor for efficient predictive maintenance, Measurement and Control, **57**, 1146–1160 (2024).

[41] Pourkiaei, Mohsen and Romain, Anne-Claude, Scoping review of indoor air quality indexes: Characterization and applications, Journal of Building Engineering, **75**, 106703 (2023).

[42] Dion, Helen and Evans, Martin and Farrell, Peter, Hospitals management transformative initiatives; towards energy efficiency and environmental sustainability in healthcare facilities, Journal of Engineering, Design and Technology, **21**, 552–584 (2023).

[43] Cowan, Kelly and Semmens, Erin O and Lee, Jeannette Y and Walker, Ethan S and Smith, Paul G and Fu, Linda and Singleton, Rosalyn and Cox, Sara McClure and Faiella, Jennifer and Chassereau, Laurie and others, Bronchiolitis recovery and the use of High Efficiency Particulate Air (HEPA) Filters (The BREATHE Study): study protocol for a multi-center, parallel, double-blind, randomized controlled clinical trial, Trials, **25**, 197 (2024).

[44] Clarke, Rachel D and Garba, Nana Aisha and Barbieri, Manuel A and Acuna, Leonardo and Baum, Marianna and Rodriguez, Maribel Saad and Frias, Hansel and Saldarriaga, Paulina and Stefano, Troy and Mathee, Kalai and others, Detection of SARS-CoV-2 in high-efficiency particulate air (HEPA) filters of low-cost air purifiers in community-based organizations, Environmental Monitoring and Assessment, **195**, 1320 (2023).

[45] Pei, Jingjing and Qu, Meinan and Sun, Luyao and Wang, Xueyong and Yin, Yihui, The relationship between indoor air quality (IAQ) and perceived air quality (PAQ)–a review and case analysis of Chinese residential environment, Energy and Built Environment, **5**, 230–243 (2024).

[46] Lu, Jie and Tian, Xiangning and Zhang, Chaobo and Zhao, Yang and Zhang, Jian and Zhang, Wenkai and Feng, Chenxin and He, Jianing and Wang, Jiaxi and He, Fengtai, Evaluation of large language models (LLMs) on the mastery of knowledge and skills in the heating, ventilation and air conditioning (HVAC) industry, Energy and Built Environment (2024).

[47] Thornton, Gail M and Fleck, Brian A and Dandnayak, Dhyey and Kroeker, Emily and Zhong, Lexuan and Hartling, Lisa, The impact of heating, ventilation and air conditioning (HVAC) design features on the transmission of viruses, including the 2019 novel coronavirus (COVID-19): A systematic review of humidity, PLoS One, **17**, e0275654 (2022).

[48] Taheri, Saman and Hosseini, Paniz and Razban, Ali, Model predictive control of heating, ventilation, and air conditioning (HVAC) systems: A state-of-the-art review, Journal of Building Engineering, **60**, 105067 (2022).

[49] Parkavi, A and Sowmya, BJ and Alex, Sini Anna and Supreeth, S and Shruthi, G and Rohith, S and Chatterjee, Sudipta and Lingaraj, K, Air quality and dust level monitoring systems in hospitals using IoT, Discover Internet of Things, **5**, 1–18 (2025).
[50] Dogan, Ahmet and Kayaci, Nurullah and Bacak, Aykut, Machine learning-based predictive model for temperature and comfort parameters in indoor enviroment using experimantal data, Applied Thermal Engineering, **259**, 124852 (2025).
[51] Bian, Jianxiao and Wang, Jiarui and Yece, Qian, A novel study on power consumption of an HVAC system using CatBoost and AdaBoost algorithms combined with the metaheuristic algorithms, Energy, **302**, 131841 (2024).
[52] Pendharkar, Parag C, Exhaustive and heuristic search approaches for learning a software defect prediction model, Engineering Applications of Artificial Intelligence, **23**, 34–40 (2010).
[53] Martino, Sergio Di and Ferrucci, Filomena and Gravino, Carmine and Sarro, Federica, A genetic algorithm to configure support vector machines for predicting fault-prone components, International conference on product focused software process improvement, 247–261 (2011).
[54] Heidari, Ali Asghar and Faris, Hossam and Aljarah, Ibrahim and Mirjalili, Seyedali, An efficient hybrid multilayer perceptron neural network with grasshopper optimization, Soft Computing, **23**, 7941–7958 (2019).
[55] Heidari, Ali Asghar and Faris, Hossam and Aljarah, Ibrahim and Mirjalili, Seyedali, An efficient hybrid multilayer perceptron neural network with grasshopper optimization, Soft Computing, **23**, 7941–7958 (2019).
[56] Ojha, Varun Kumar and Abraham, Ajith and Snas, el, Vaclav, Metaheuristic design of feedforward neural networks: A review of two decades of research, Engineering Applications of Artificial Intelligence, **60**, 97–116 (2017).
[57] Yamany, Waleed and Fawzy, Mohammed and Tharwat, Alaa and Hassanien, Aboul Ella, Moth-flame optimization for training multi-layer perceptrons, 2015 11th International computer engineering Conference (ICENCO), 267–272 (2015).
[58] Shatnawi, Raed, The application of ROC analysis in threshold identification, data imbalance and metrics selection for software fault prediction, Innovations in Systems and Software Engineering, **13**, 201–217 (2017).
[59] Erturk, Ezgi and Sezer, Ebru Akcapinar, A comparison of some soft computing methods for software fault prediction, Expert systems with applications, **42**, 1872–1879 (2015).
[60] Mirjalili, Seyedali and Gandomi, Amir H and Mirjalili, Seyedeh Zahra and Saremi, Shahrzad and Faris, Hossam and Mirjalili, Seyed Mohammad, Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems, Advances in engineering software, **114**, 163–191 (2017).
[61] Qiao, Lei and Li, Xuesong and Umer, Qasim and Guo, Ping, Deep learning based software defect prediction, Neurocomputing, **385**, 100–110 (2020).
[62] Abusnaina, Ahmed A and Ahmad, Sobhi and Jarrar, Radi and Mafarja, Majdi, Training neural networks using salp swarm algorithm for pattern classification, Proceedings of the 2nd International Conference on Future Networks and Distributed Systems, 17 (2018).
[63] Rao, R Venkata and Saroj, Ankit, An elitism-based self-adaptive multi-population Jaya algorithm and its applications, Soft Computing, **23**, 4383–4406 (2019).
[64] Shen, Xin and Wu, Guohua and Wang, Rui and Chen, Huangke and Li, Haifeng and Shi, Jianmai, A self-adapted across neighborhood search algorithm with variable reduction strategy for solving non-convex static and dynamic economic dispatch problems, IEEE Access, **6**, 41314–41324 (2018).
[65] Wang, Hui and Sun, Hui and Li, Changhe and Rahnamayan, Shahryar and Pan, Jeng-shyang, Diversity enhanced particle swarm optimization with neighborhood search, Information Sciences, **223**, 119-135 (2013).
[66] Raut, Usharani and Mishra, Sivkumar, An improved Elitist–Jaya algorithm for simultaneous network reconfiguration and DG allocation in power distribution systems, Renewable Energy Focus, **30**, 92–106 (2019).
[67] Hassouneh, Yousef and Turabieh, Hamza and Thaher, Thaer and Tumar, Iyad and Chantar, Hamouda and Too, Jingwei, Boosted whale optimization algorithm with natural selection operators for software fault prediction, IEEE Access, **9**, 14239–14258 (2021).
[68] AG, Priya Varshini and Varadarajan, Vijayakumar and others, Estimating software development efforts using a random forest-based stacked ensemble approach, Electronics, **10**, 1195 (2021).
[69] Sharma, Sudhir and Vijayvargiya, Shripal, An optimized neuro-fuzzy network for software project effort estimation, IETE Journal of Research, 1–12 (2022).
[70] Kaushik, Anupama and Singal, Niyati, A hybrid model of wavelet neural network and metaheuristic algorithm for software development effort estimation, International Journal of Information Technology, 1–10 (2019).
[71] Kumari, Sweta and Pushkar, Shashank, Cuckoo search based hybrid models for improving the accuracy of software effort estimation, Microsystem Technologies, **24**, 4767–4774 (2018).
[72] Keung, Jacky and Kocaguneli, Ekrem and Menzies, Tim, Finding conclusion stability for selecting the best effort predictor in software effort estimation, Automated Software Engineering, **20**, 543–567 (2013).
[73] Ali, Asad and Gravino, Carmine, Improving software effort estimation using bio-inspired algorithms to select relevant features: An empirical study, Science of Computer Programming, **205**, 102621 (2021).
[74] Rjoub, Gaith and Elmekki, Hanae and Bentahar, Jamal and Pedrycz, Witold and Kassaymeh, Sofian and Almobydeen, Shahed Bassam and Dssouli, Rachida, Enhanced Dynamic Deep Q-Network for Federated Learning scheduling policies on IoT devices using explanation-driven trust, Knowledge-Based Systems, 113574 (2025).
[75] Makhadmeh, Sharif Naser and Awadallah, Mohammed A and Kassaymeh, Sofian and Al-Betar, Mohammed Azmi and Sanjalawe, Yousef and Kouka, Shaimaa and Al-Redhaei, Anessa, Recent advances in Multi-objective Cuckoo Search Algorithm, its variants and applications, Archives of Computational Methods in Engineering, 1–28 (2025).
[76] Saraireh, Jameel and Agoy, Mary and Kassaymeh, Sofian, Adaptive Ensemble Learning Model-Based Binary White Shark Optimizer for Software Defect Classification, International Journal of Computational Intelligence Systems, **18**, 1–51 (2025).
[77] Kassaymeh, Sofian and Al-Betar, Mohammed Azmi and Rjoubd, Gaith and Fraihat, Salam and Abdullah,

Salwani and Almasri, Ammar, Optimizing beyond boundaries: empowering the salp swarm algorithm for global optimization and defective software module classification, Neural Computing and Applications, **36**, 18727–18759 (2024).
[78] Awadallah, Mohammed A and Braik, Malik and AlAkhras, Leen and Kassaymeh, Sofian and Al-Betar, Mohammed Azmi, White Shark Optimizer and its Applications: A Systematic Review, Archives of Computational Methods in Engineering, 1–69 (2025).
[79] Kassaymeh, Sofian and Alweshah, Mohammed and Al-Betar, Mohammed Azmi and Hammouri, Abdelaziz I and Al-Maaitah, Mohammad Atwah, Software effort estimation modeling and fully connected artificial neural network optimization using soft computing techniques, Cluster Computing, 1–24 (2023).
[80] Kassaymeh, Sofian and Abdullah, Salwani and Al-Laham, Mohamad and Alah, Mohammed and Al-Betar, Mohammed Azmi and Othman, Zalinda, Salp Swarm Optimizer for Modeling Software Reliability Prediction Problems, Neural Processing Letters, 1–37 (2021).
[81] Al-Betar, Mohammed Azmi and Kassaymeh, Sofian and Makhadmeh, Sharif Naser and Fraihat, Salam and Abdullah, Salwani, Feedforward neural network-based augmented salp swarm optimizer for accurate software development cost forecasting, Applied Soft Computing, **149**, 111008 (2023).
[82] Alweshah, Mohammed and Almiani, Muder and Alkhalaileh, Saleh and Kassaymeh, Sofian and Hezzam, Essa Abdullah and Alomoush, Waleed, Parallel Metaheuristic Algorithms for Solving Imbalanced Data Classification Problems, IEEE Access (2023).
[83] Sharma, Vibhu and Mistry, Vrushank, Machine learning algorithms for predictive maintenance in HVAC systems, Journal of Scientific and Engineering Research, **10** (2023).

Bilal Bataineh received the B.S in computer science and information system from Philadelphia University Amman, Jordan in 1998, and M.S degrees from Red Sea University, Al-Khartoum, Sudan in 2002 and Ph.D. degree in Computer Information System / Artificial intelligence from Arab Academy for Banking and Financial Sciences Amman, Jordan in 2008. He is an Assistant Professor at the Faculty of Information Technology, Department of Computer Science, Jadara University, Irbid, Jordan. His current research interests include Artificial Intelligence, machine learning, security, Image Processing and NLP.