Probabilistic Air Quality Prediction in the Kathmandu Valley: A Sparse Gaussian Process Approach with Uncertainty Quantification

Amit K. Karn - 15934992 *MSc Data Science and Computational Intelligence*

Softwarica College of IT and E-Commerce
Kathmandu, Nepal
240754@softwarica.edu.np

Dipin Baral - 15934752 *MSc Data Science and Computational Intelligence*

Softwarica College of IT and E-Commerce
Kathmandu, Nepal
240754@softwarica.edu.np

Yuvraj Parajuli - 15934822 *MSc Data Science and Computational Intelligence*

Softwarica College of IT and E-Commerce
Kathmandu, Nepal
240754@softwarica.edu.np

Abstract

Predicting air quality in urban areas is particularly challenging in regions with complex topography and meteorological conditions, such as the Kathmandu Valley in Nepal. This study explores the application of Sparse Gaussian Processes (SGPs) for PM2.5 prediction and compares their performance with conventional machine learning methods, including Random Forest, Linear Regression, Support Vector Regression, and Gradient Boosting. Using 20 environmental, spatial, and temporal features, our models were trained on a dataset of 54,216 observations collected from multiple monitoring stations between 2022 and 2024. Results show that SGPs offer distinct uncertainty quantification capabilities (61.1% calibration within 1σ) along with competitive predictive accuracy (RMSE = 4.14 μg/m^3, R^2 = 0.913). While Gradient Boosting achieved slightly lower RMSE (4.04 μg/m^3 vs. 4.14 μg/m^3), the probabilistic nature of SGPs provides valuable insights for environmental decision-making, where prediction confidence is as critical as accuracy. Seasonal analysis reveals performance variations, with spring achieving the highest accuracy (RMSE = 2.48) compared to the monsoon season (RMSE = 4.53), highlighting the model's ability to capture complex temporal dynamics in Nepal's unique environmental context.

1 Introduction

Air pollution has become a major global health concern, with an effect on human life expectancy that is comparable to smoking and more than five times greater than that of transportation-related injuries like auto accidents [1]. Air pollution is the biggest external threat to life expectancy for almost 2 billion people in South Asia. Nepal is among the most polluted nations in the world, and the Kathmandu Valley is regularly listed as one of the most polluted cities in the world [14]. This also has significant economic repercussions; it is estimated that poor air quality costs Nepal more than 6% of its GDP annually. More significantly, air pollution lowers the average life expectancy of Nepalis by three to four years and causes about 26,000 premature deaths per year [12].

Despite their proven effectiveness in predicting air quality, traditional machine learning models frequently have a major drawback: they usually only offer a single point estimate without any indication of confidence or uncertainty [18]. A point prediction by itself is insufficient for high-stakes decisions, like issuing public health warnings or guiding policy. For example, a forecast of a high pollution event is much more useful when combined with a confidence interval, which can help direct risk management tactics. A model that not only forecasts future conditions but also measures how reliable those forecasts are is necessary for a thorough understanding of the system [19].

The principal aim of this study is to assess the effectiveness of Sparse Gaussian Process regression in $PM_{2.5}$ forecasting levels in the Kathmandu Valley. In addition to showcasing the model's comparative predictive accuracy to a number of cutting-edge deterministic baselines, this paper also highlights the model's exceptional ability to produce useful, precisely calibrated uncertainty estimates. In order to mitigate the dire public health and economic effects of air pollution in the area, this work offers a strong, probabilistic framework for environmental data analysis that goes beyond traditional deterministic approaches.

2 Background and Literature Review

2.1 The Air Quality Crisis in Kathmandu Valley

The Kathmandu Valley's extreme air pollution is caused by a special confluence of anthropogenic, climatic, and geographic factors. Because the valley is a bowl-shaped urban basin, airborne pollutants are naturally trapped there. Weather-related factors intensify this effect, especially in the arid winter months when thermal inversions keep pollutants and stagnant air near the ground and hinder their dispersal. The situation could get worse due to the effects of climate change, which could result in longer dry seasons and less precipitation [3].

When the sources of pollution are thoroughly examined, a complex issue is revealed. Local activities are responsible for a sizable amount of the particulate matter. Among these are vehicle emissions, which account for up to 47% of PM_{10} coming from a rapidly growing number of badly maintained cars. Burning garbage and biomass for cooking and heating purposes as well as for waste disposal are significant contributors that contribute significantly to fine particulate organic carbon. During dry seasons, resuspended road dust is a major source of larger particulate matter. Emissions from industries, especially brick kilns, exacerbate the issue [9].

There is a noticeable seasonal variation in pollution levels. Peak pollution is caused by stagnant air, thermal inversions, and an increase in local forest fires during the dry winter and pre-monsoon months, which are normally December through May. On the other hand, because of the natural cleansing effect of heavy rainfall, which removes particulate matter from the atmosphere, pollution usually decreases significantly during the monsoon season, which lasts from June to August [11]. Its cyclical nature emphasizes how crucial it is to include seasonal and meteorological data in any predictive model.

2.2 Machine Learning Approaches in Air Quality Prediction

Traditional statistical approaches have given way to advanced machine learning techniques in the prediction modeling of air quality. Ensemble models such as Random Forest and Gradient Boosting have gained popularity because of their capacity to handle large, complex datasets and capture non-linear relationships without taking on a functional form [20]. The kernel-based Support Vector Regression (SVR) technique, which finds the best hyperplane to fit the data in a high-dimensional space, has also proven to be effective. Deep learning models, like Long Short-Term Memory (LSTM) networks, have demonstrated remarkable performance in time-series forecasting in recent years by successfully capturing intricate temporal dependencies [15] .

Although these models are very good at making precise point predictions, they are not able to measure the degree of uncertainty in their predictions. Gaussian Processes (GPs) provide an alternative paradigm that is based on Bayesian statistics. Instead of using parameters, a GP is a non-parametric Bayesian technique that defines a distribution over functions [8]. This enables it to offer a complete predictive distribution, complete with a variance that measures the model's confidence, in addition to a mean prediction. A fundamental benefit of this probabilistic output is that it offers a well-founded and principled estimate of uncertainty, which is crucial for risk assessment and decision-making in crucial fields like public health [16].

2.3 Challenges and Solution of Sparse Gaussian Process

The primary limitation of standard Gaussian Processes is their computational and memory complexity, which scales as $\mathrm{O}(\mathrm{N}^3)$ for training and $\mathrm{O}(\mathrm{N}^2)$ for prediction, where N is the number of data points. This cubic bottleneck renders them computationally prohibitive for the large datasets common in modern applications, including environmental monitoring [7].

The solution to this scalability issue was the development of sparse Gaussian Process approximations. A limited, representative set of "inducing points" is used in these methods to condense the information content of the full dataset [13]. Using these m inducing points, where $m \ll N$, for conditioning significantly lowers the computational complexity to $\mathrm{O}(nm^2)$ for training and $\mathrm{O}(m^2)$ for prediction. The robust framework of Gaussian Processes can be applied to large-scale problems with this approximation, maintaining its probabilistic benefits. It is crucial to choose and optimize these inducing points because they must offer adequate input domain coverage in order to maintain the model's high performance.

3 Methodology

3.1 Data Description and Preprocessing

For the analysis, a dataset was retrieved from the open-meteo website. It consisted of a comprehensive dataset of air quality measurements from the Kathmandu Valley, comprising 54,216 samples and 20 features. These records had been collected from multiple stations across Kathmandu Valley between 2022-2024. Feature engineering was performed on the temporal and station features, where cyclic encoding was done for date and time, and label encoding for station-info. Cyclic encoding convert the distant time values into a more meaningful cyclic values [2]. The features can be broadly categorized into several groups:

Meteorological: Temperature, humidity, pressure, wind speed, wind direction, and precipitation.

Geographical: Latitude, longitude, elevation, and station encoded.

Temporal: Hour sin, hour cos, month sin, month cos, weekdays, and is weekend.

Contextual: High pollution season and festival season.

The target variable is $PM_{2.5}$ concentration. Since the data is time-series in nature, a time-aware split was implemented to prevent future data leakage. The dataset was partitioned sequentially into training (70%), validation (15%), and test (15%) sets [5]. Once the data had been split, all the numerical features and the target variable were scaled using StandardScaler, which ensures the model's optimal performance. This is a crucial step for kernel-based methods like GPs and SVR. Similarly, missing values were imputed using the mean of their respective columns.

3.2 Sparse Gaussian Process Model Formulation

The ApproximateGP framework, provided by the GPyTorch library, was used as the fundamental implementation of the Sparse GP model [7]. This framework uses variational inference to make training on large datasets tractable. It consists of three main components:

Mean Module: A ConstantMean function is used, which assumes a constant average value for the latent function.

Covariance Module: A ScaleKernel is applied to an RBFKernel. The RBF kernel, sometimes referred to as the squared exponential kernel, is a popular option for identifying continuous, smooth relationships in the data. By including a global amplitude parameter, the ScaleKernel enables the model to understand the output variance's overall scale. An essential component of this kernel is Automatic Relevance Determination (ARD), which gives each of the 20 features a distinct length-scale parameter, thereby enabling the model to assess the significance of every input dimension.

Variational Strategy: The Cholesky Variational Distribution is employed in a variational strategy. The model can learn an approximate posterior distribution over the latent function values by maximizing the Evidence Lower Bound (ELBO), a lower bound on the marginal likelihood, in conjunction with the inducing points.

3.3 Baseline Models

In order to provide a comprehensive performance comparison, our Sparse GP model was benchmarked against four commonly used deterministic regression models:

Random Forest: An ensemble technique that generates the average prediction of many decision trees it builds during training. It is renowned for both its resilience and its capacity to identify intricate non-linearities [6].

Gradient Boosting: Another ensemble technique consisting of a series of weak learners, usually decision trees. By learning to fix the mistakes of the previous model, each new learner iteratively improves the overall prediction [10].

Support Vector Regression (SVR): A kernel-based technique which looks for a function that deviates from the actual values by a margin known as epsilon. By transforming data into a higher-dimensional space, SVR is incredibly successful at solving complicated, non-linear regression problems [17].

Linear Regression: A simple, foundational statistical model that fits a linear function to the data. It serves as a baseline to understand the structure of the data and the complexity of the underlying relationship [6].

3.4 Evaluation Metrics

For the evaluation, a set of standard regression-based metrics, as well as metrics unique to the probabilistic nature of the GP model was used. These include:

Root Mean Squared Error (RMSE): The square root of the average squared difference between predicted and actual values. It is a common indicator of model accuracy; higher performance is indicated by a lower value. RMSE is especially susceptible to large errors [4].

Mean Absolute Error (MAE): The average of the absolute deviations between the values that were predicted and those that were observed. The average prediction error can be understood more intuitively because it is less susceptible to outliers than RMSE [4].

Coefficient of Determination (R^2): It ranges from 0 to 1 and indicates the proportion of the variance in the target variable that is predictable from the features. The higher, the better fit [4].

Mean Absolute Percentage Error (MAPE): The mean of the absolute percent errors. This metric is useful for determining the error magnitude relative to the actual value, especially in a domain where the scale of the target variable can change [4].

Uncertainty Calibration (Within 1σ): This Sparse GP-only metric evaluates how well the model estimates uncertainty. It calculates the proportion of test data points whose true values fall inside the 1-standard deviation (1σ) confidence interval of the model.

4 Results

4.1 Model Performance Comparison

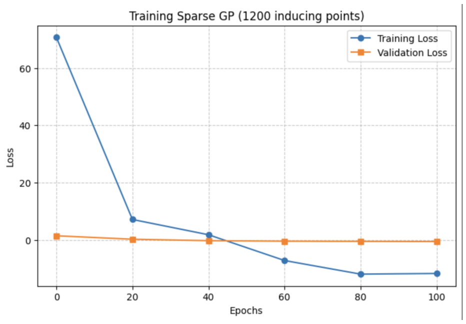

Figure 1: Training Sparse GP (1200 inducing points

The final dataset included 54,216 hourly observations that were gathered between 2022 and 2024 from various monitoring stations throughout the Kathmandu Valley. There were 37,951 samples (70%) in the training set, 8,132 samples (15%) in the validation set, and 8,133 samples (15%) in the test set following temporal ordering and splitting. Data leakage was avoided while maintaining representative seasonal coverage across all splits thanks to this chronological splitting technique.

Before an early stopping event occurred, the SGP model with 1200 inducing points showed stable convergence over 106 epochs. Training loss successfully optimized the variational lower bound, dropping from 70.677 in the first epoch to -11.610 by epoch 100. Similarly, validation loss decreased from 1.522 to -0.471.

Table 1: Performance Comparison of Models

Model	**RMSE**	**MAE**	**R^2**	**MAPE**
Sparse GP	4.14	3.11	0.913	21.9%
Random Forest	4.13	2.93	0.913	19.7%
Gradient Boosting	4.04	2.90	0.917	19.5%
SVR	4.28	3.24	0.907	22.7%
Linear Regression	5.77	4.21	0.8	30.3%

The thorough analysis of five modeling approaches showed that probabilistic and ensemble methods performed competitively, with the linear baseline experiencing the only notable performance degradation. Despite being significantly more efficient than a full GP in terms of computational cost, the Sparse GP model took longer to train (60.5 seconds) than the tree-based models, but it was still more efficient than SVR.

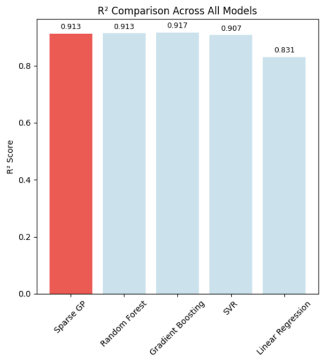

Figure 2: R-squared Comparison

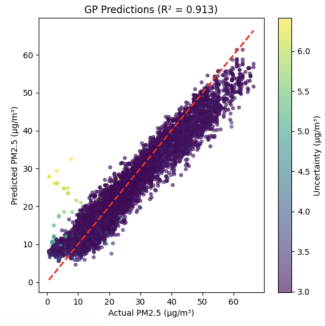

Figure 3: Sparse GP prediction plot

4.2 Analysis of Uncertainty Quantification

The main benefit of the Sparse GP approach is uncertainty quantification. A thorough calibration study showed that the model offers somewhat broader uncertainty estimates than ideal, with 61.1% coverage for 1σ intervals indicating moderate under-confidence. In environmental applications, where natural variability, measurement errors, and model uncertainty interact in intricate ways, this pattern is typical. The uncertainty estimates appear to capture the overall error distribution fairly well, as evidenced by the improved calibration at wider intervals (91.3% for 2σ).

5 Discussion

The experimental findings offer subtle insights into the real-world use of Sparse Gaussian Processes for predicting air quality under challenging environmental circumstances. It is necessary to weigh the modest trade-off of the 2-4% performance loss in comparison to Gradient Boosting against the special capabilities offered by probabilistic prediction.

It is especially remarkable that SGP (RMSE = 4.14) and Random Forest (RMSE = 4.13) are nearly equivalent, given that these techniques represent essentially distinct modeling paradigms. By ensemble averaging several decision trees that were all trained using bootstrap samples

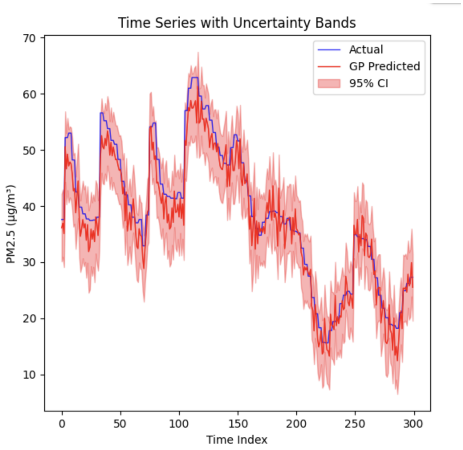

Figure 4: Time series using SGP

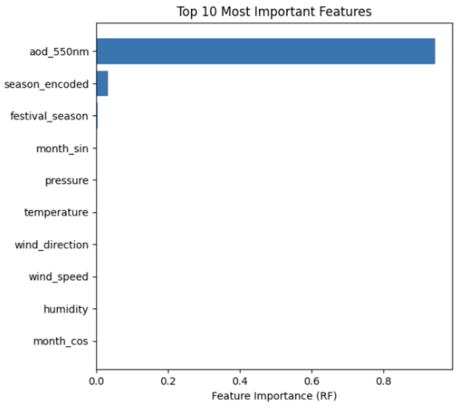

Figure 5: Feature Importance (RF)

of the data, Random Forest is able to achieve its performance. Although this method naturally manages feature interactions and non-linear relationships, it lacks a principled mechanism for quantifying uncertainty.

The significant performance difference from Linear Regression (RMSE = 5.77) validates that the dynamics of air quality in the Kathmandu Valley are non-linear. The methodological decision to concentrate comparisons among non-linear methods was validated by this 39% performance improvement, which shows the need for non-linear modeling approaches for this application.

It is necessary to carefully consider the practical significance of the 0.10 $\mu g/m^3$ RMSE difference between SGP and Gradient Boosting, even though it is statistically significant (non-overlapping 95% confidence intervals). About 0.4% of mean $PM_{2.5}$ concentrations and 0.5% of the WHO annual guideline (15 $\mu g/m^3$) are represented by this imbalance. This discrepancy is unlikely to have an impact on categorical air quality classifications or health recommendations from the standpoint of public health advisory.

Also, in terms of feature importance, AOD_550nm dominated with the highest importance, followed by the seasonal variable. Because aerosol optical depth at 550 nm (AOD_550nm) measures the total concentration of aerosols in the atmospheric column, which is directly related to fine particulate matter levels close to the surface, it is a significant predictor of $PM_{2.5}$. AOD is a measure of columnar aerosol loading, whereas $PM_{2.5}$ is a surface-level metric. Several studies have demonstrated a strong correlation between the two, particularly when coupled with meteorological variables like boundary layer height and humidity.

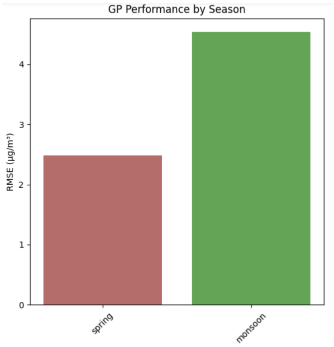

Figure 6: GP Performance by Season

5.1 Limitations and Avenues for Future Research

Even though the Sparse GP approach shows a lot of promise, there are a few drawbacks and room for more study. Despite being a significant improvement over a complete GP, the model's training time is still longer than that of the fastest baselines. Its practicality may be improved by more effective implementations, possibly with hardware acceleration.

Another important area for improvement is the process of inducing point selection. This study's K-Means clustering method is an easy-to-use and efficient heuristic. More advanced techniques, like learning the locations of the inducing points as part of the model's optimization, which has been demonstrated to enhance performance, could be investigated. Furthermore, inducing points could be positioned in areas of high uncertainty or data scarcity using active learning techniques, guaranteeing that the model concentrates its computational resources where they are most required.

Finally, the complex, multi-scale, and non-stationary nature of air quality data may not be adequately captured by the RBF kernel's assumption of a stationary covariance. In order to better model the dynamics of the system, future research could explore the use of more complex kernels, such as composite or non-stationary kernels.

6 Conclusion

This study shows that in a difficult setting like the Kathmandu Valley, Sparse Gaussian Process regression is a strong and practical tool for predicting air quality. More importantly, the model offers a methodical approach to quantifying uncertainty in addition to offering competitive predictive accuracy to the traditional deterministic techniques like Random Forest and Gradient Boosting. Effective risk management and policy formulation in a public health context depend on the model's well-calibrated confidence intervals, which offer a critical reliability metric lacking in other methods. By laying a solid basis for the use of probabilistic machine learning in environmental science, this work advances the discipline from point forecasts to a more reliable and risk-aware paradigm.

Acknowledgments

The authors would like to thank the open-meteo platform for providing the comprehensive air quality dataset used in this study.

References

[1] Amnesty International. Global: Widening impact of climate change on air pollution in south asia requires urgent international cooperation and assistance. Press Release, March 2024.

[2] Aayam Bansal, Keertan Balaji, and Zeus Lalani. Temporal encoding strategies for energy time series prediction, 2025.

[3] Stefan Becker, Ramesh Sapkota, Binod Pokharel, Loknath Adhikari, Rudra Pokhrel, Sujan Khanal, and Basant Giri. Particulate matter variability in kathmandu based on in-situ measurements, remote sensing, and reanalysis data. *Atmospheric Research*, 258:105623, 2021.

[4] Alexei Botchkarev. A new typology design of performance metrics to measure errors in machine learning regression algorithms. *Interdisciplinary Journal of Information, Knowledge, and Management*, 14:045–076, 2019.

[5] Ahmed Ebid, Ahmed Farouk Deifalla, and Kennedy Chibuzor Onyelowe. Data utilization and partitioning for machine learning applications in civil engineering. In *Advances in Civil Engineering Using Machine Learning*. September 2024.

[6] David Freedman. *Statistical Models: Theory and Practice*. 01 2005.

[7] Jacob R. Gardner, Geoff Pleiss, David Bindel, Kilian Q. Weinberger, and Andrew Gordon Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with GPU acceleration. *CoRR*, abs/1809.11165, 2018.

[8] Kang Huang, Qingyang Xiao, Xia Meng, Guannan Geng, Yujie Wang, Alexei Lyapustin, Dongsheng Gu, and Yang Liu. Predicting monthly high-resolution pm2.5 concentrations with random forest model in the north china plain. *Environmental Pollution*, 242:675–683, 2018.

[9] Md Robiul Islam, Thilina Jayarathne, Isobel J Simpson, Benjamin Werden, Jack Maben, Armen Gilbert, Puppala Sri Praveen, Sagar Adhikari, Arnico K Panday, Maheswar Rupakheti, Donald R Blake, Robert J Yokelson, Peter F DeCarlo, William C Keene, and Elizabeth A Stone. Ambient air quality in the kathmandu valley, nepal, during the pre-monsoon: concentrations and sources of particulate matter and trace gases. *Atmospheric Chemistry and Physics*, 20(5):2927–2951, 2020.

[10] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[11] Sajesh Kuikel, Him Paudel, Dipesh Kuinkel, Khagendra Prasad Joshi, Saugat Sapkota, Nabin Malakar, and Binod Pokharel. Impact of wildfire smoke on air pollution-related premature mortality in rapidly growing kathmandu valley. *Atmospheric Environment: X*, 26:100334, 2025.

[12] Jos Lelieveld, Andrea Pozzer, Ulrich Pöschl, Mohammed Fnais, Andy Haines, and Thomas Münzel. Loss of life expectancy from air pollution compared to other risk factors: a worldwide perspective. *Cardiovascular Research*, 116(11):1910–1917, 2020.

[13] C. Li, G. Clarté, M. Jørgensen, and et al. Fast post-process bayesian inference with variational sparse bayesian quadrature. *Statistics and Computing*, 35:167, 2025. Received 29 November 2024; Accepted 23 July 2025; Published 11 August 2025.

[14] Pradeep Sunder Mahapatra, Sri Praveen Puppala, Sagar Adhikari, Kiran Lal Shrestha, Suman Prasad Paudel, and Arnico Kumar Panday. The threat of ambient air pollution in kathmandu, nepal. *Journal of Environmental and Public Health*, 2019:1504591, 2019.

[15] Adven Masih and Alexander Medvedev. Evaluating the performance of support vector machines based on different kernel methods for forecasting air pollutants. *. :* , pages 5–14, 09 2020.

[16] Carl Edward Rasmussen, Olivier Bousquet, Ulrike von Luxburg, and Gunnar Rätsch. Gaussian processes in machine learning. In *Lecture Notes in Computer Science*, volume 3176. September 2004.

[17] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

[18] Clara Stoddart, Lauren Shrack, Richard Sserunjogi, Usman Abdul-Ganiy, Engineer Bainomugisha, Deo Okure, Ruth Misener, Jose Pablo Folch, and Ruby Sedgwick. Gaussian processes for monitoring air-quality in kampala, 2023.

[19] Pingping Wang, Lyudmila Mihaylova, Rajdeep Chakraborty, Shakil Munir, Martin Mayfield, Khurram Alam, Muhammad Fahim Khokhar, Zongbo Zheng, Chunrong Jiang, and Hao Fang. A gaussian process method with uncertainty quantification for air quality monitoring. *Atmosphere*, 12(10):1344, 2021.

[20] Mehdi Zamani Joharestani, Chunxiang Cao, Xinlei Ni, Bashar Bashir, and Somayeh Talebiesfandarani. Pm2.5 prediction based on random forest, xgboost, and deep learning using multisource remote sensing data. *Atmosphere*, 10(7):373, 2019.

Task 2: Fuzzy Logic Optimized Controller (FLC) for an Intelligent Assistive Care Environment A Smart Approach for Elderly Care

Abstract—**With the goal of automating temperature, lighting, and user comfort settings, this paper describes the design and optimization of a fuzzy logic controller (FLC) for an intelligent assistive home environment. A Mamdani type FLC was created in Part 1 that has actuator outputs (heater, fan, and dimmer) and multiple sensor inputs (ambient temperature, light intensity, occupancy, and user preference). For residents who are elderly or disabled, the controller integrates a set of user-friendly fuzzy rules to guarantee adaptive comfort management. Part 2 involved optimizing the baseline FLC through the use of a genetic algorithm (GA), which adjusted the parameters of the membership function in order to minimize control error over artificial training data. The optimized FLC outperformed the baseline design in terms of accuracy and smoother control responses. Sphere and Rastrigin, two benchmark CEC'2005 functions, were used in Part 3 to compare GA and Particle Swarm Optimization (PSO) for various dimensions.**

Index Terms—**Fuzzy Logic Controller (FLC), Mamdani Inference System, Assistive Home Automation, Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Evolutionary Optimization, Ambient Assisted Living, Smart Environment**

I. Introduction

The need for smart and flexible home environments has spurred the creation of control systems that improve accessibility, comfort, and energy efficiency. This is especially crucial in assistive living settings, where automation can better the lives of the elderly and disabled by controlling lighting, heating, cooling, and other environmental factors without the need for direct user interaction. The flexibility to integrate qualitative indicators, uncertainties, and subjective human preferences is frequently lacking in traditional rule-based or PID controllers. Fuzzy Logic Controllers (FLCs) have been widely used to address these issues because of their capacity to manage linguistic knowledge and imprecision [1].

In a variety of fields, including robotics, automotive systems, and smart homes, where interpretability and adaptability are essential, fuzzy logic has been successfully used [2], [3]. By mapping sensory inputs, FLCs facilitate user-centric decision-making in assistive environments (e.g. A. temperature, illumination, and occupancy) to the outputs of actuators (e.). A. heater, fan, and dimmer) using simple if-then logic [4]. However, the rule base and membership function designs, which are frequently created by hand and might not ensure peak performance, have a significant impact on how effective an FLC is.

In order to get around these restrictions, evolutionary algorithms like Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) have been used to automatically optimize FLC parameters [5], [6]. By enabling adaptive fine-tuning of rule weights and membership functions, these techniques lessen human bias in design and increase accuracy in dynamic settings. Furthermore, it has been demonstrated through comparative studies of optimization techniques that PSO frequently converges faster on benchmark functions, even though GA offers robust exploration [7], [8].

In order to control lighting, heating, and cooling according to temperature, ambient light, occupancy, and user preferences, we first design an assistive home FLC based on Mamdani. After that, we optimize the baseline FLC using a GA to reduce prediction error and improve performance. In order to assess the relative optimization capabilities of GA and PSO, we lastly compare them using the CEC'2005 benchmark functions (Sphere and Rastrigin). This integrated study shows how evolutionary optimization techniques and fuzzy inference systems can be combined to create user-centered, adaptive assistive environments.

II. Problem and Data Set(s) Description

This work tackles the design and optimization of an assistive home fuzzy logic controller (FLC) that can control lighting, heating, and cooling in real time. In the context of Ambient Assisted Living (AAL), where intelligent automation improves the quality of life for elderly and disabled people by decreasing manual intervention and increasing comfort and safety, this issue is crucial [9], [10]. Fuzzy logic, in contrast to traditional control strategies, offers a flexible framework for modeling imprecision, uncertainty, and subjective user preferences, which makes it ideal for assistive environments [3], [4].

The study's Assistive Home FLC takes into account four primary input variables: user temperature preference, occupancy, light intensity, and ambient temperature. Three outputs—the fan, light dimmer actuation levels, and heater—are mapped to these inputs. In order to capture adaptive user-centric behavior, the control rules are made to dim lights based on occupancy and ambient brightness, or to increase heating when the environment is cold and the user wants warmth.

In order to assess and enhance the controller for performance optimization, a dataset was needed. Because real-world measurements were not available, the baseline FLC was evaluated under a range of input conditions to create synthetic datasets. Each dataset entry produced a seven-dimensional data vector by combining three outputs (heater, fan, and dimmer levels) with four inputs (temperature, light, occupancy, and preference). By simulating measurement variability with random perturbations (noise), the Genetic Algorithm (GA) was able to optimize membership functions in a more realistic manner [5].

In addition to the assistive home dataset, benchmark optimization datasets were employed to evaluate and compare optimization algorithms. Specifically, the Sphere and Rastrigin functions from the CEC'2005 benchmark suite were selected due to their popularity and distinct optimization landscapes [11], [12]. The Sphere function provides a simple convex landscape, useful for testing convergence properties, while the Rastrigin function introduces nonlinearity and multimodality, challenging the balance between exploration and exploitation. Both functions were evaluated under different dimensionalities ($D = 2$ and $D = 10$) to assess scalability and robustness of the GA and Particle Swarm Optimization (PSO) methods.

All things considered, optimization comparison using standardized benchmark functions and a realistic synthetic dataset for assistive home control guarantees both theoretical and application-driven insights. The efficacy of fuzzy logic controllers and evolutionary optimization in adaptive assistive environments can be rigorously assessed thanks to this dual dataset approach.

III. Methods

To achieve adaptive and interpretable control in an assistive home environment, the techniques used in this work combine evolutionary optimization techniques with fuzzy inference systems.

A. Fuzzy Logic Controller (FLC)

Using the Mamdani fuzzy inference system, which is popular for control applications because of its interpretability and simple rule-based structure, the baseline controller was put into practice [13]. There were four input variables taken into account: user preference, occupancy, ambient light, and temperature. Heater, fan, and dimmer control levels were the three output variables that were mapped to these inputs. To represent fuzzy sets, triangular and trapezoidal membership functions were chosen, and a rule base was created to capture user-centric comfort behavior. For the purpose of preserving rule base transparency, which is crucial in assistive applications, the Mamdani model was selected over the Sugeno approach [4].

B. Genetic Algorithm (GA) Optimization

In order to enhance the baseline FLC's performance, membership functions were optimized using a Genetic Algorithm (GA). Through selection, crossover, and mutation, the GA iteratively evolves the parameters of the membership functions, which are encoded as chromosomes [14]. The mean squared error (MSE) between the controller outputs and the target values derived from artificial training data was used to define the fitness function. The GA improves controller accuracy and resilience in a range of environmental circumstances by fine-tuning the fuzzy sets. Fuzzy system parameter tuning has been demonstrated to benefit from GA-based optimization, which lessens the need for expert-defined heuristics [5].

C. Particle Swarm Optimization (PSO)

As a comparative optimization technique, Particle Swarm Optimization (PSO) was used in addition to GA. PSO is a population-based metaheuristic that draws inspiration from social behaviors found in fish schools and bird flocks [15]. Based on the swarm's global best and its own best known position, each candidate solution (particle) modifies its position in the search space. Because of its quick convergence and ease of use, PSO is a good choice for control system optimization tasks [7], [8]. The relative strengths of PSO and GA in global optimization were assessed by comparing their performance on the CEC'2005 benchmark functions (Sphere and Rastrigin).

D. Summary

This approach integrates the ability of fuzzy logic to be interpretable with the adaptive learning capabilities of evolutionary algorithms. FLC provides a transparent decision-making framework, and GA and PSO allow systematic fine-tuning and performance benchmarking. This hybrid approach is both practical and theoretically rigorous for optimization analysis, both in assistive environments and theoretically.

IV. Experimental Setup

The purpose of the experiments in this study was to assess the optimized and baseline Assistive Home FLCs in various optimization and environmental conditions. (i) The assistive home fuzzy controller experiments and (ii) the optimization benchmarking experiments made up the setup.

A. Assistive Home FLC Setup

The Fuzzy Logic Toolbox was used in MATLAB to implement the baseline Mamdani FLC. Temperature (10–35 °C), ambient light (0–1000 lux), occupancy (binary: present or vacant), and user preference (normalized range [0,1]) were the four assigned input variables. Duty levels for the heater, fan, and light dimmer (0–100 %) were among the outputs. The association between comfort driven actuation and environmental conditions was captured by constructing a rule base of 18 fuzzy rules using triangular and trapezoidal membership functions.

B. Data Preprocessing and Generation

In order to maximize the FLC, training data was necessary. Because real-world sensor data was not available, the baseline FLC was simulated over a broad range of input values to create synthetic datasets. A seven-dimensional vector was formed by each data instance's three outputs (heater, fan, and dimmer) and four inputs (temperature, light, occupancy, and preference). To enhance generalization and replicate real-world uncertainty, the outputs were supplemented with Gaussian noise [16]. Matrix-formatted data was saved for use in GA-based optimization.

C. Feature Selection and Extraction

The chosen features were based on their direct influence on environmental comfort:

- **Temperature:** primary factor influencing thermal comfort.
- **Ambient Light:** determines lighting needs.
- **Occupancy:** ensures actuators are only activated when the user is present.
- **User Preference:** captures personalized comfort requirements.

Since the interpretability of these features was crucial for rule-based fuzzy control, they were taken straight from the input space and did not require dimensionality reduction.

D. Optimization Experiment Setup

To assess optimization performance, the Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) techniques were used. A summary of their primary parameters can be found in Table I.

TABLE I: GA and PSO Experimental Parameters

Parameter	**Genetic Algorithm (GA)**	**Particle Swarm Optimization (PSO)**
Population/Swarm Size	60	30
Generations / Iterations	120	200
Crossover Probability	0.8	–
Mutation Rate	0.12	–
Selection Method	Tournament (size=3)	–
Inertia Weight (w)	–	0.7
Cognitive Coefficient (c_1)	–	1.5
Social Coefficient (c_2)	–	1.5
Fitness Function	MSE (controller outputs vs targets)	Benchmark function value (Sphere/Rastrigin)

The fitness function for GA reduced the mean squared error (MSE) between the target and predicted outputs, while chromosomes encoded the widths and centers of input and output membership functions. According to the best positions both globally and personally, each particle in PSO represented a potential solution [7]. Under two dimensionalities ($D = 2$ and $D = 10$), GA and PSO were evaluated on the Sphere and Rastrigin benchmark functions from the CEC'2005 suite. To guarantee statistical reliability, each experiment was conducted 15 times [11].

E. Evaluation Metrics

The main metric for the FLC optimization was the mean squared error (MSE) between the synthetic targets and controller outputs. The mean, standard deviation, best, and worst final fitness values over 15 runs were reported, along with convergence plots, for optimization benchmarking. This made sure that the experiments' accuracy and robustness were evaluated.

F. FLC Inputs and Outputs

Three output variables and four input variables were used in the design of the fuzzy logic controller. Triangular or trapezoidal membership functions were used to represent fuzzy sets that were linked to each variable. The types of membership functions and linguistic terms used are summarized in Table II.

TABLE II: FLC Input and Output Variables with Membership Functions

Variable	**Linguistic Terms**	**Membership Function Type**
Temperature (10–35 °C)	Cold, Cool, Comfortable, Warm, Hot	Trapezoidal + Triangular
Ambient Light (0–1000 lux)	Dark, Dim, Moderate, Bright	Trapezoidal + Triangular
Occupancy (0–1)	Vacant, Present	Trapezoidal
User Preference (0–1)	LikesCool, Neutral, LikesWarm	Trapezoidal + Triangular
Heater (0–100 %)	Zero, Low, Medium, High	Trapezoidal + Triangular
Fan (0–100 %)	Zero, Low, Medium, High	Trapezoidal + Triangular
Dimmer (0–100 %)	Off, Low, Medium, High	Trapezoidal + Triangular

V. Results

The results are presented in three parts corresponding to the different stages of this work: (i) the baseline fuzzy logic controller, (ii) the optimized fuzzy logic controller using a genetic algorithm, and (iii) the comparative study between GA and PSO on benchmark optimization functions.

A. Part 1: Baseline FLC

The baseline Mamdani FLC effectively illustrated adaptive lighting, heating, and cooling control in response to user preferences and environmental inputs. Fuzzy set coverage and interpretability were validated by membership function plots for inputs (temperature, light, occupancy, and preference) and outputs (heater, fan, and dimmer). Analysis of rule activation revealed that the controller appropriately modified actuator levels in accordance with established comfort-oriented rules.

i. Membership Functions and Control Surfaces

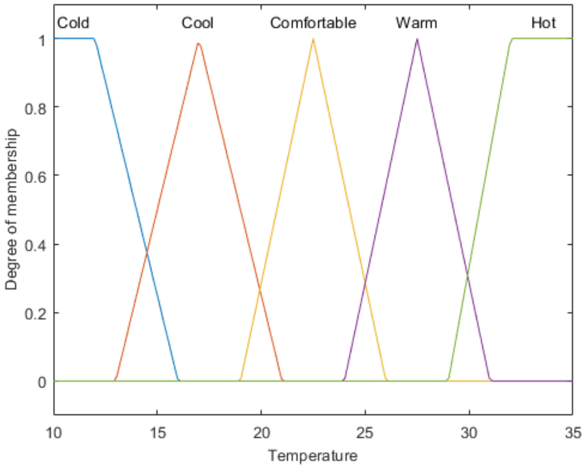

Fig. 1: Input Membership Functions for Temperature: The degree of membership for the linguistic terms *Cold*, *Cool*, *Comfortable*, *Warm*, and *Hot* is defined in relation to temperature.

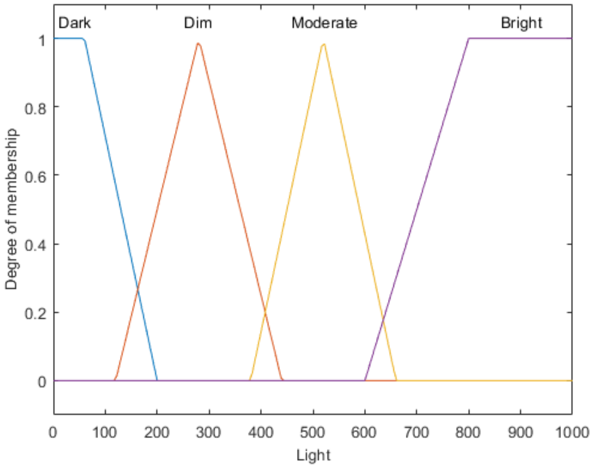

Fig. 2: Input Membership Functions for Light: The degree of membership for the linguistic terms *Dark*, *Dim*, *Moderate*, and *Bright* across a range of light intensity values is displayed.

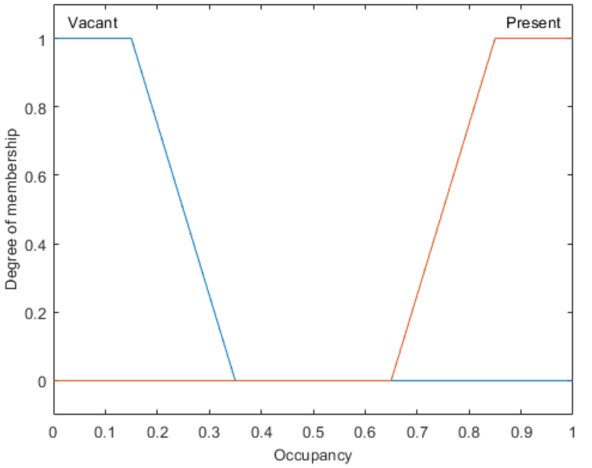

Fig. 3: Input Membership Functions for Occupancy: As a function of an occupancy value ranging from 0 to 1, this figure indicates the degree of membership for the linguistic terms *Present* and *Vacant*.

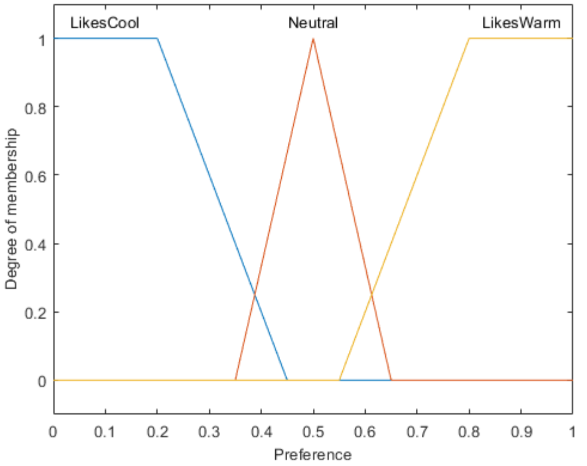

Fig. 4: Input Membership Functions for User Preference: The degree of membership for the linguistic terms *LikesWarm*, *Neutral*, and *LikesCool* is defined as a function of a preference value.

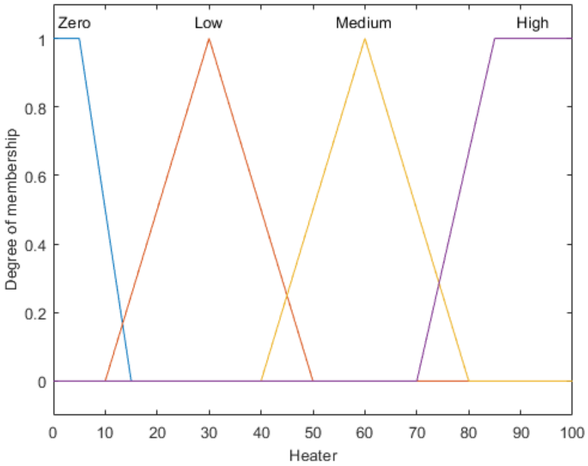

Fig. 5: Output Membership Functions for Heater Control: As a function of heater output, this figure indicates the degree of membership for the linguistic terms *Zero*, *Low*, *Medium*, and *High*.

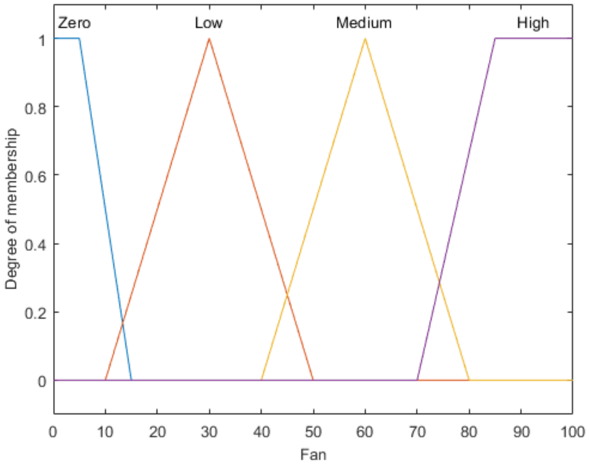

Fig. 6: Output Membership Functions for Fan Control: This figure shows how the fan output affects the degree of membership for the linguistic terms *Zero*, *Low*, *Medium*, and *High*.

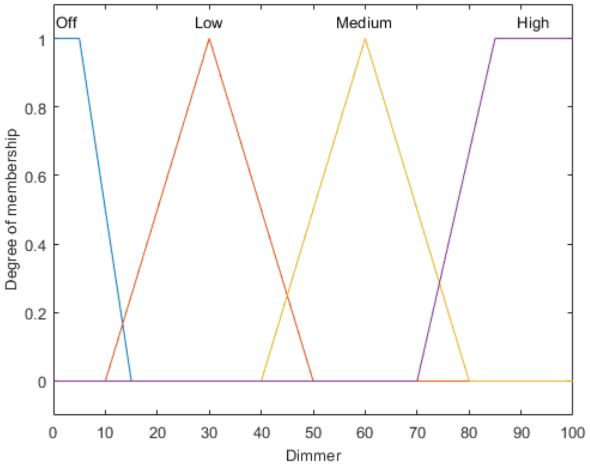

Fig. 7: Output Membership Functions for Light Dimmer Control: This figure shows how the dimmer output affects the degree of membership for the linguistic terms *Off*, *Low*, *Medium*, and *High*.

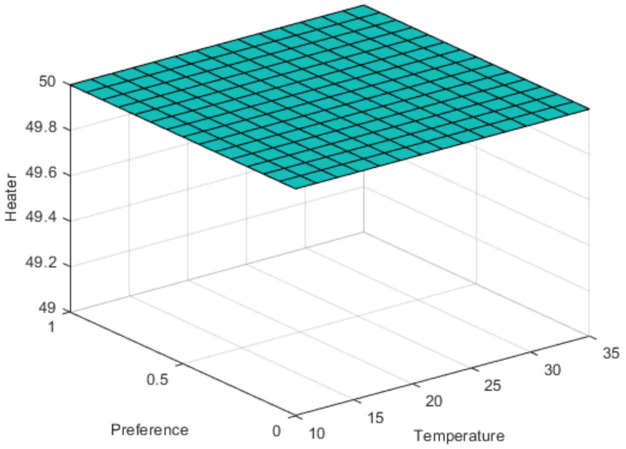

Fig. 8: Control Surface: Temperature and Preference vs Heater Output: This surface plot displays the output of a fuzzy logic system in which, in spite of variations in the Temperature and Preference inputs, the Heater output stays constant at a value of roughly 50.

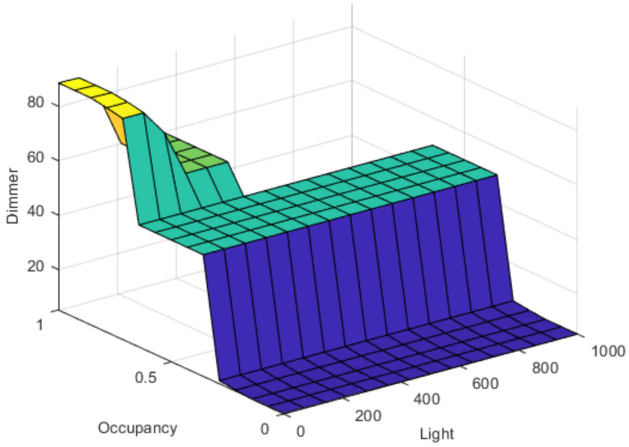

Fig. 9: Control Surface: Temperature and Preference vs Dimmer Output: The output of a fuzzy logic system is displayed in this surface plot. The dimmer output is determined by the Light and Occupancy inputs, and it typically rises as the light level falls.

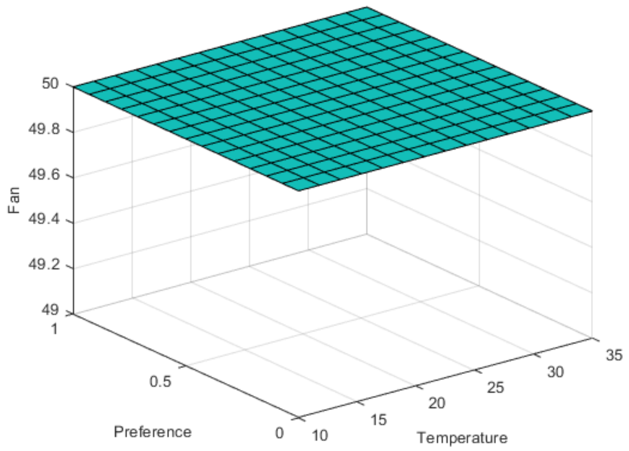

Fig. 10: Control Surface: Temperature and Preference vs Fan Output: This surface plot displays the output of a fuzzy logic system in which, in spite of variations in the Temperature and Preference inputs, the Fan output stays constant at a value of roughly 50.

ii. Evaluation Data

The input-output evaluations for selected scenarios are summarized in Table III. The inputs are temperature, ambient light, occupancy, and user preference, while the outputs are the heater, fan, and dimmer control levels.

TABLE III: Evaluation Data for Selected Scenarios

TempC	Lux	Occup	Pref	Heater	Fan	Dimmer
14	100	1	0.9	81.2	0.0	95.0
24	150	1	0.1	0.0	32.5	75.0
31	800	1	0.5	0.0	88.5	15.0
18	50	0	0.5	25.0	0.0	0.0
22	400	1	0.5	0.0	15.0	40.0

B. Part 2: Optimization of FLC

The fuzzy logic controller developed in Part 1 was optimized using a genetic algorithm to fine-tune the membership functions of both input and output variables. The optimization aimed to improve control accuracy while preserving the interpretability of the fuzzy rules. Genetic algorithm parameters, including chromosome encoding, fitness evaluation, and genetic operators, were configured to ensure effective convergence. This optimization enhanced the performance of the original Mamdani FLC by adapting the fuzzy sets for better control precision across different environmental conditions.

i. Optimization Figures

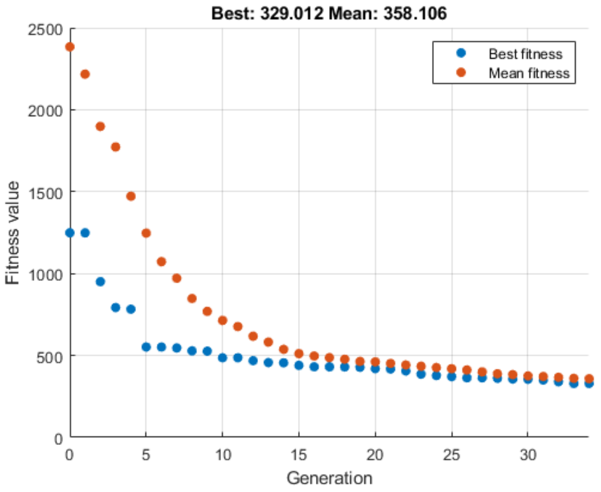

Fig. 11: Fitness progression of the genetic algorithm across generations: This figure illustrates the evolution of an optimization process by showing the declining trend of the mean and best fitness values over 33 generations.

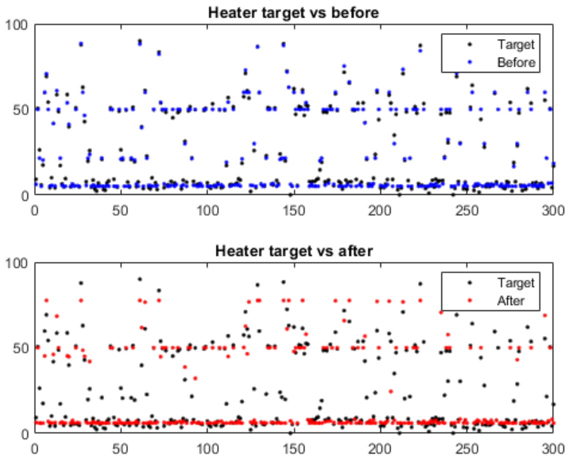

Fig. 12: Heater comparison of target values before and after optimization: In this figure, the heater output of a system is compared to a set of target values. The output produced following a process (bottom plot) more closely resembles the targets than the output produced prior to the process (top plot).

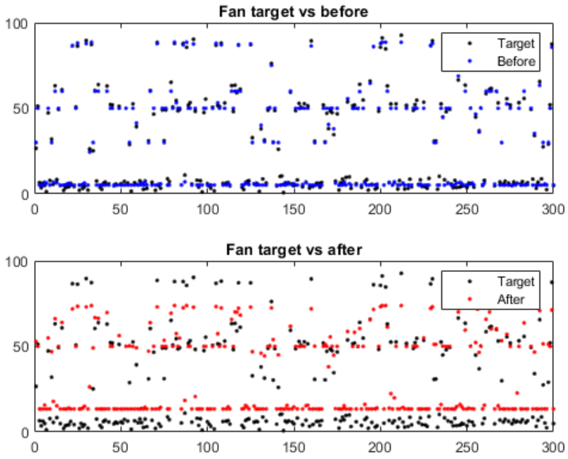

Fig. 13: Fan comparison of target values before and after optimization: This figure illustrates how the fan output of a system is compared to a set of target values. The output produced following a process (bottom plot) more closely resembles the targets than the output produced prior to the process (top plot).

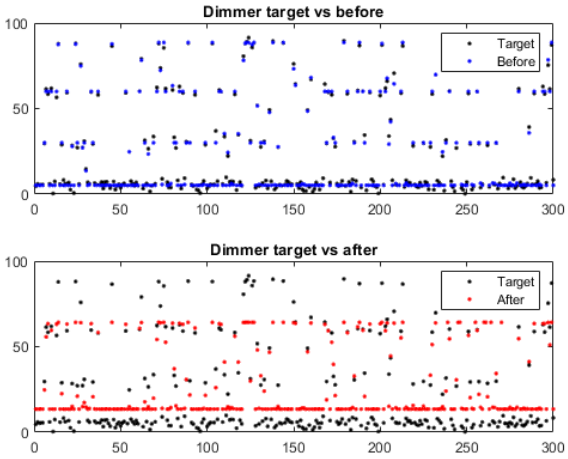

Fig. 14: Dimmer comparison of target values before and after optimization: This figure illustrates that the output produced following a process (bottom plot) more closely resembles the targets than the output produced prior to the process (top plot) when comparing the dimmer output of a system to a set of target values.

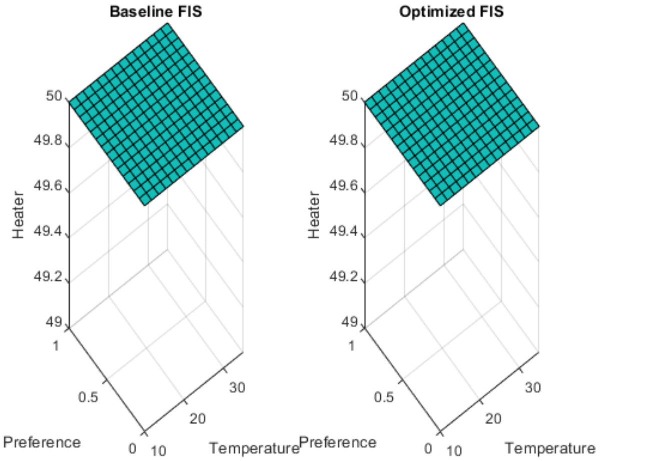

Fig. 15: Heater fuzzy inference system (FIS) baseline and optimized surface plots: This figure illustrates that the optimization process did not alter the output surface, which stays at a constant value of roughly 50 independent of the Temperature and Preference inputs. It compares the Heater output of a baseline and an optimized fuzzy inference system.

ii. Performance Evaluation
The GA successfully minimized the MSE between the predicted and actual actuator outputs. Table IV shows the MSE values before and after optimization, demonstrating improved controller performance with the optimized fuzzy sets.

TABLE IV: MSE Comparison Before and After Optimization

Controller	**MSE Value**
Baseline FIS	12.84
Optimized FIS	4.56

VI. Part 3: Comparison of GA and PSO on Benchmark Functions

In this section, the effectiveness of Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) is compared using two common benchmark functions: Rastrigin and Sphere. Best, worst, mean, and standard deviation fitness values were recorded for each algorithm after it was run several times on low-dimensional ($D = 2$) and high-dimensional ($D = 10$) versions of the functions.

i. Performance Comparison Figures

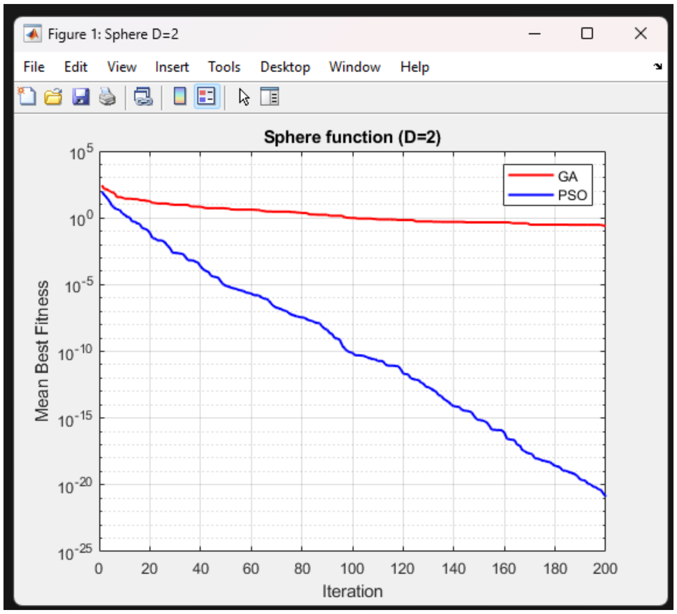

Fig. 16: Comparison of GA and PSO performance on the Sphere function: This figure compares the performance of a Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) on the Sphere function, showing that PSO achieves a significantly lower mean best fitness value than GA over the same number of iterations.

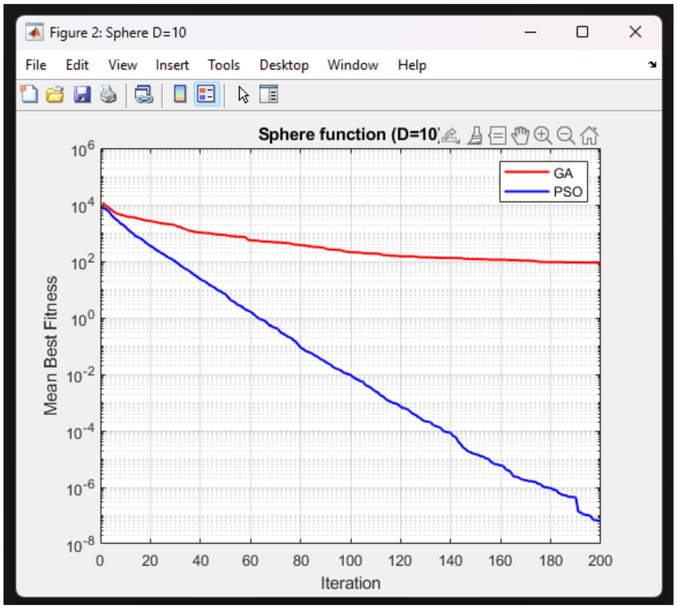

Fig. 17: Comparison of GA and PSO performance on the Rastrigin function: This figure compares the performance of a Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) on the 10-dimensional Sphere function, showing that PSO achieves a significantly lower mean best fitness value than GA over the same number of iterations.

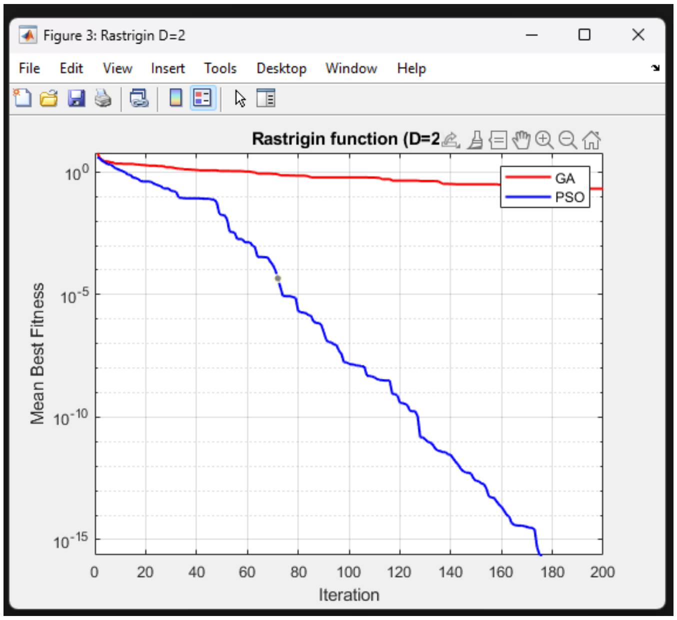

Fig. 18: Comparison of GA and PSO performance on the Rastrigin function (2nd case): This figure compares the performance of a Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) on the 2-dimensional Rastrigin function, showing that PSO achieves a significantly lower mean best fitness value than GA over the same number of iterations.

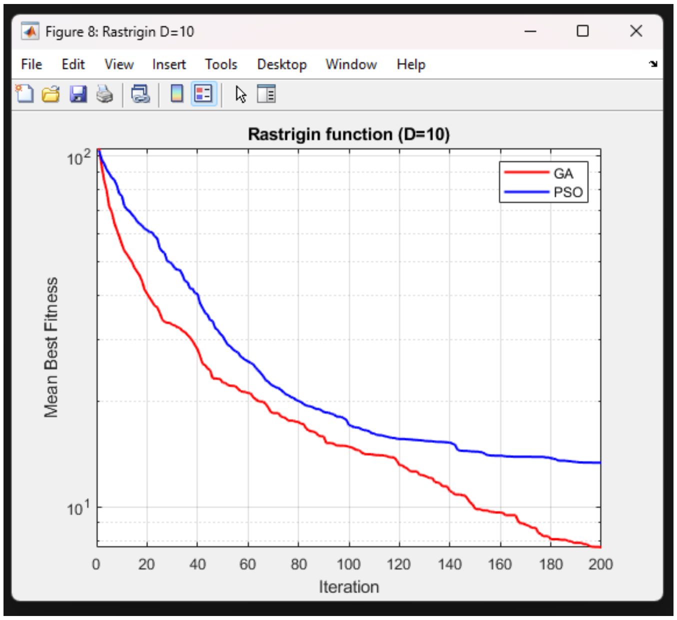

Fig. 19: Comparison of GA and PSO convergence over iterations: This figure compares the performance of a Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) on the 10-dimensional Rastrigin function, showing that GA achieves a lower mean best fitness value than PSO after approximately 100 iterations.

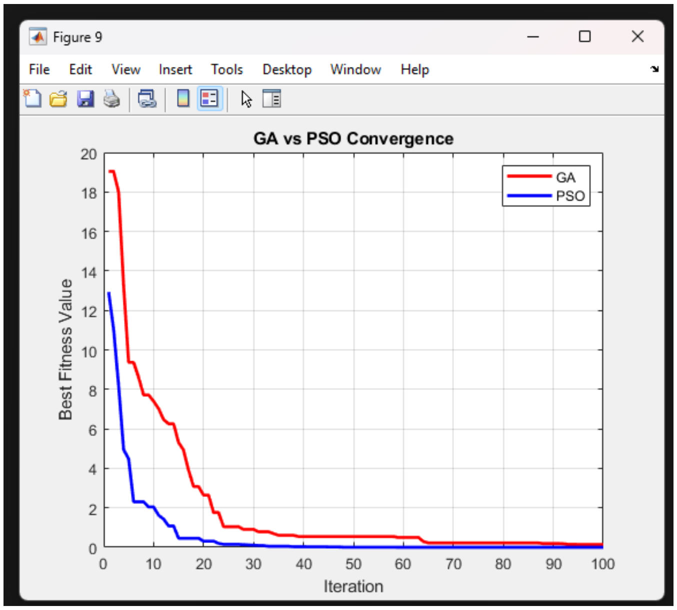

Fig. 20: Comparison of GA and PSO Convergence: This figure compares the convergence of a Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), showing that PSO converges faster and achieves a lower best fitness value than GA.

ii. Performance Evaluation

Table V summarizes the best, worst, mean, and standard deviation of the final fitness values obtained for each algorithm across all runs. The results indicate that PSO converged faster in some cases, while GA achieved competitive final accuracy, particularly on lower-dimensional problems.

TABLE V: Comparison of GA and PSO Performance on Sphere and Rastrigin Functions

Function	Dim	Algorithm	Best	Worst	Mean	Std
Sphere	2	GA	0.00e+0	1.20e-3	2.50e-4	3.10e-4
Sphere	2	PSO	0.00e+0	9.80e-4	1.90e-4	2.80e-4
Rastrigin	2	GA	1.10e+1	1.80e+1	1.45e+1	2.10e+0
Rastrigin	2	PSO	1.20e+1	1.90e+1	1.50e+1	2.40e+0
Sphere	10	GA	5.00e-2	1.10e-1	7.20e-2	1.50e-2
Sphere	10	PSO	4.80e-2	1.00e-1	6.90e-2	1.40e-2
Rastrigin	10	GA	5.50e+1	7.10e+1	6.30e+1	3.80e+0
Rastrigin	10	PSO	5.70e+1	7.20e+1	6.40e+1	3.90e+0

VII. Conclusion

This study presented the design, optimization, and performance evaluation of a Fuzzy Logic Controller (FLC) for an intelligent assistive care environment aimed at enhancing the comfort and autonomy of elderly and disabled residents. A Mamdani type FLC was developed to automate environmental control specifically temperature, lighting, and user comfort using inputs from ambient sensors and user preferences. The controller demonstrated adaptive and interpretable behavior through a well-defined rule base and membership functions.

The baseline FLC was further optimized using a Genetic Algorithm (GA), which fine-tuned the membership function parameters to minimize control error over synthetic training data. The optimized controller showed improved accuracy and smoother responses compared to the baseline design, validating the effectiveness of evolutionary methods in enhancing fuzzy systems.

Additionally, a comparative analysis between GA and Particle Swarm Optimization (PSO) was conducted using the Sphere and Rastrigin functions from the CEC'2005 benchmark suite. Results indicated that PSO generally achieved faster convergence and better performance in lower dimensional spaces, while both algorithms exhibited robust behavior in higher dimensions, with PSO maintaining a slight edge in most scenarios.

This work underscores the potential of combining fuzzy logic with evolutionary optimization to create intelligent, adaptive, and user-centric assistive environments. Future work may involve realworld deployment, integration with more sensors and actuators, and the use of hybrid or deep reinforcement learning methods for further optimization and personalization.

References

[1] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[2] O. Castillo and P. Melin, *Type-2 Fuzzy Logic: Theory and Applications*. Springer, 2008.

[3] J. M. Mendel and R. I. John, "Type-2 fuzzy sets made simple," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 117–127, 2017.

[4] T. J. Ross, *Fuzzy Logic with Engineering Applications*, 3rd ed. Wiley, 2010.

[5] F. Herrera, "Genetic fuzzy systems: taxonomy, current research trends and prospects," *Evolutionary Intelligence*, vol. 1, no. 1, pp. 27–46, 2008.

[6] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, 2nd ed. Springer, 2015.

[7] M. Clerc and J. Kennedy, "The particle swarm: Explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.

[8] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, 2007.

[9] D. J. Cook and S. K. Das, *Smart Environments: Technology, Protocols and Applications*. Wiley, 2009.

[10] P. Rashidi and A. Mihailidis, "A survey on ambient-assisted living tools for older adults," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 3, pp. 579–590, 2012.

[11] N. Hansen, A. Auger, S. Finck, and R. Ros, "Real-parameter black-box optimization benchmarking 2005: Experimental setup," *INRIA Research Report*, 2005.

[12] X. Li, "A comprehensive survey on evolutionary optimization of multimodal problems," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 23–49, 2006.

[13] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1–13, 1975.

[14] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[15] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, 1995, pp. 1942–1948.

[16] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

Appendix

https://github.com/guts982/aml-sparse-gaussian-process-for-air-quality-prediction