ISSN: 3023 - 8544

# ADBA COMPUTER SCIENCE

**VOLUME 3, ISSUE 1, JANUARY 2026**

AN INTERDISCIPLINARY JOURNAL OF COMPUTER SCIENCE

https://journals.**adbascientific**.com/acs

ISSN: 3023-8544

ADBA Computer Science
Volume: 3 – Issue No: 1 (January 2026)

# EDITORIAL BOARD



https://journals.adbascientific.com/acs/about/editorialTeam

ISSN: 3023-8544

ADBA Computer Science
Volume: 3 – Issue No: 1 (January 2026)

# CONTENTS



https://journals.adbascientific.com/acs/issue/view/18

ADBA Computer Science
Cutting-Edge Scientific Solutions

e-ISSN: 3023-8544
RESEARCH ARTICLE
Vol.3 / No.1 / 2026 / pp.1-5
https://doi.org/10.69882/adba.cs.2026011

# Determination of Optimal Linear Congruence Generators Parameters with Heuristic Optimization Algorithms

**Sinem Akyol** *,1 **and Fatih Özkaynak** *,α,2

*Fırat University Department of Software Engineering Elazığ, Türkiye, αKriptarium R&D Software Consulting Defense Industry and Trade Ltd. Co., 23119, Elazığ, Türkiye.

**ABSTRACT** Random number generators play a critical role in many scientific and engineering applications, particularly in simulation, cryptography, and optimization. Among these generators, Linear Congruential Generators (LCGs) are widely used due to their simplicity and computational efficiency. However, the statistical quality of the generated sequences strongly depends on the proper selection of generator parameters. In this study, the determination of optimal LCG parameters is formulated as a heuristic optimization problem. The Improved Grey Wolf Optimizer (IGWO) is employed to search for suitable multiplier, increment, and seed values. The proposed approach aims to achieve a uniform distribution of generated numbers while maintaining low correlation between consecutive values. The performance of the optimized LCG is evaluated using the objective fitness function as well as additional statistical performance metrics derived from the generated sequences. The effectiveness of the proposed IGWO-based optimization approach is demonstrated through repeated independent runs using the same fitness evaluation framework. Experimental results demonstrate that the proposed approach provides improved parameter selection for LCGs and enhances the statistical properties of the generated random sequences.

**KEYWORDS**
Linear congruential generator
Random number generation
Heuristic optimization
Grey wolf optimizer
Statistical randomness

## INTRODUCTION

Random number generation constitutes a fundamental component of modern scientific computing and engineering applications (Bikos *et al.* 2023). Pseudo-random number generators (PRNGs) are extensively used in Monte Carlo simulations, numerical integration, stochastic differential equations, optimization algorithms, and uncertainty quantification frameworks (Zhao *et al.* 2023; Ozkaynak and Yavuz 2013; Park *et al.* 2022). In these domains, the statistical quality of generated random sequences directly influences the accuracy, stability, and reproducibility of computational results (Zhao *et al.* 2023; Zhao and Ma 2024). Consequently, the design, analysis, and evaluation of reliable random number generators remain an active and critical area of research. Monte Carlo methods, in particular, rely heavily on high-quality pseudo-random sequences to approximate solutions to problems that are analytically intractable (Li *et al.* 2024; Irfan *et al.* 2020). The convergence rate and variance of Monte Carlo estimators are strongly affected by the uniformity and independence properties of the underlying random numbers (Foreman *et al.* 2024).

Deficiencies in randomness may introduce systematic bias, slow convergence, or misleading confidence intervals, ultimately compromising the validity of simulation-based studies (Ferreira *et al.* 2023). Similar concerns arise in numerical methods where random sampling is employed, such as randomized algorithms for linear algebra, probabilistic numerical schemes, and sensitivity analyses (Foreman *et al.* 2024; Ferreira *et al.* 2023; Álvarez *et al.* 2022). Random number generation also plays a significant role in machine learning and stochastic optimization techniques. Algorithms such as stochastic gradient descent, evolutionary computation, particle swarm optimization, and reinforcement learning depend on randomness for exploration, parameter initialization, and probabilistic decision-making (Maksymovych *et al.* 2022).

Poor statistical properties may lead to premature convergence, reduced diversity, or unstable learning dynamics, especially in large-scale or high-dimensional problems. While randomness is also essential in cryptographic applications, the requirements for cryptographic security differ fundamentally from those of simulation-oriented randomness (Maksymovych *et al.* 2022). Cryptographically secure random number generators demand resis-

**Manuscript received:** 1 December 2025,
**Revised:** 15 January 2026,
**Accepted:** 19 January 2026.

[1]sakyol@firat.edu.tr
[2]ozkaynak@firat.edu.tr(**Corresponding author**).

tance against adversarial prediction and reconstruction, whereas simulation-focused PRNGs prioritize statistical quality, reproducibility, and computational efficiency (Ozkaynak 2014; Nannipieri *et al.* 2021; Crocetti *et al.* 2022). As emphasized by Knuth, Marsaglia, and L'Ecuyer, a generator that is suitable for simulation purposes may be entirely inadequate for cryptographic use, and vice versa (Ozkaynak 2020; Rojas-Muñoz *et al.* 2022). Therefore, it is essential to clearly distinguish between these application domains when evaluating and optimizing random number generators (Liu *et al.* 2021). Extensive research has demonstrated that many widely used PRNGs exhibit hidden structural weaknesses that may only become apparent under rigorous statistical testing.

Marsaglia's Diehard tests and L'Ecuyer's TestU01 framework have revealed that generators passing basic theoretical criteria can still fail advanced statistical evaluations (Nam *et al.* 2022; Hajduk 2024). These findings underscore the necessity of systematic parameter selection and empirical validation, particularly for generators intended for large-scale simulations and numerical experiments (Almaraz Luengo and Román Villaizán 2023). In this context, improving the statistical reliability of simulation-oriented random number generators remains a crucial challenge (Ozkaynak 2015). Rather than relying solely on classical parameter choices or theoretical constraints, modern approaches increasingly emphasize empirical performance evaluated through comprehensive test suites. In these contexts, the quality of pseudo-random sequences impacts both the efficiency of the optimization process and the robustness of learned models. Therefore, the design and evaluation of random number generators remain an active research topic.

Linear Congruential Generators (LCGs) are among the simplest and most commonly used pseudo-random number generators due to their ease of implementation and low computational cost (Steele Jr. and Vigna 2022). An LCG generates a sequence of numbers based on a recurrence relation defined by a multiplier, an increment, a modulus, and an initial seed value (Steele Jr. and Vigna 2022). Despite their simplicity, LCGs are known to suffer from statistical weaknesses when their parameters are not carefully selected (L'Ecuyer 1999). Poor parameter choices may result in short periods, non-uniform distributions, and strong correlations between successive numbers. Although LCGs have been extensively studied in the literature, many existing works primarily focus on theoretical conditions for achieving maximum period lengths or optimize parameters using a single statistical criterion (Panda and Ray 2020). However, optimizing LCG parameters based solely on one performance measure may lead to suboptimal random behavior when other statistical properties are taken into account. In practice, the quality of a random number generator should be evaluated using multiple criteria, such as uniformity and independence, simultaneously. In this study, the parameter selection problem of LCGs is formulated as a heuristic optimization problem. The Improved Grey Wolf Optimizer (IGWO) is employed to search for suitable LCG parameters. The main contributions of this work can be summarized as follows:

- The formulation of the LCG parameter selection problem as a heuristic optimization task.
- The application of the Improved Grey Wolf Optimizer (IGWO) to determine suitable generator parameters.
- The evaluation of the optimization performance through multiple independent runs using a consistent fitness function.

Due to the scope of the present study, the analysis is limited to a single heuristic optimization method and a fitness-based evaluation, while broader comparative and multi-metric analyses are left for future investigations. The remainder of this paper is organized as follows. Section 2 introduces the LCG model, the proposed optimization framework and the experimental results, and Section 3 concludes the paper.

## LCG MODEL AND THE PROPOSED OPTIMIZATION FRAMEWORK

Heuristic optimization algorithms are a class of methods inspired by natural systems, designed to solve complex problems by incorporating heuristic knowledge into the problem-solving process. This section explains the fundamental principles of heuristic optimization algorithms and their application in optimizing random number generators.

Intelligent heuristic optimization algorithms are widely used to solve complex real-world problems due to their simplicity and high performance (Akyol 2022). Many of these problems can be modeled as optimization problems, and heuristic methods can be quickly adapted as solution-seeking strategies Qiu *et al.* (2024). Heuristic optimization algorithms are developed based on inspiration from various disciplines, including genetic algorithms, particle swarm optimization, simulated annealing, and harmony search (Yu *et al.* 2024). Their core principle is to work iteratively on a population of candidate solutions in the problem domain to find the best solution. These algorithms evaluate the current solutions at each step and use these evaluations to create new generations of solutions (Chen and Zheng 2024). The process concludes with the maximization or minimization of a specific criterion or objective function (Rajwar *et al.* 2023). These algorithms are classified into nine categories based on their sources of inspiration: music-based, mathematics-based, swarm-based, social-based, biology-based, chemistry-based, sports-based, physics-based, and hybrid methods combining these approaches (Akyol and Alatas 2017). Among these, biology-based and swarm-based metaheuristic methods are frequently preferred for search and optimization problems. In this study, an improved version of the Grey Wolf Optimizer, a swarm-based heuristic algorithm inspired by the hunting strategies of grey wolves, has been utilized (Mirjalili *et al.* 2014).

Heuristic optimization algorithms are commonly used in optimizing random number generators. Specifically, they can be effectively employed to determine and improve the parameters of random number generators such as LCGs. Heuristic optimization algorithms optimize an objective function, which, in the case of LCG parameter optimization, typically evaluates the statistical properties of the generated random numbers. The objective function aims to maximize uniform distribution, minimize serial correlation, and achieve the desired period of random numbers. Heuristic optimization algorithms explore the parameter space and use an iterative approach to find the best parameter combination. This process is crucial to ensure that the random number generator possesses the desired statistical properties.

In this study, the Improved Grey Wolf Optimizer (IGWO) proposed by Nadimi-Shahraki et al. Nadimi-Shahraki *et al.* (2021) was used to determine LCG configurations with good statistical properties. IGWO is a swarm-based metaheuristic method inspired by the hunting skills and leadership hierarchies within packs of grey wolves. In the IGWO method, which draws on the hunting behavior and natural social leadership of grey wolves, there are four types of wolves. The wolves in the best positions are categorized as $\alpha$, $\beta$, and $\delta$ wolves, while the remaining $\omega$ wolves are guided by these leaders. The hunting process of the wolves consists of seven stages: encircling, hunting, attacking prey, initiating, moving, selecting, and updating (Mirjalili *et al.* 2014). The hunting strategy is

illustrated in Figure 1.

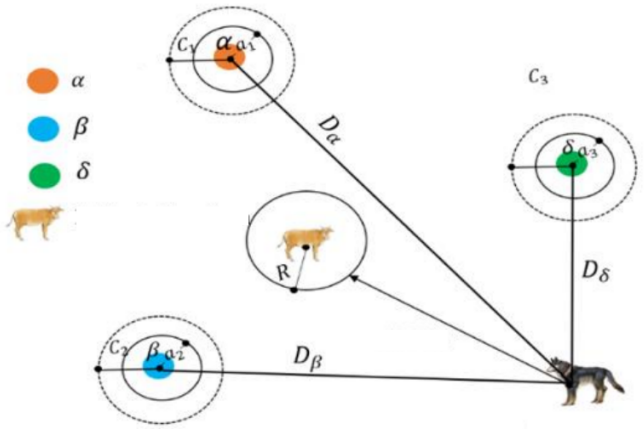



**Figure 1** Hunting strategies of gray wolves

**Encirclement:** This is the phase where the prey is surrounded by grey wolves and is expressed by Eq. (1) and Eq. (2).

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)| \tag{1}$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \tag{2}$$

Where $\vec{X}_p$ represents the location of the prey, $t$ represents the current iteration, $\vec{X}$ represents the location vector of the gray wolf, and the coefficients $\vec{A}$ and $\vec{C}$ are calculated with the equations in Eq. (3) and Eq. (4) (Mirjalili *et al.* 2014).

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \tag{3}$$

$$\vec{C} = 2 \cdot \vec{r}_2 \tag{4}$$

Each time the algorithm is run, the element values of the $\vec{a}$ vector are linearly reduced from 2 to 0. $\vec{r}_1$ and $\vec{r}_2$ are random vectors that take values in the range $[0, 1]$.

**Hunting:** In this stage, which mathematically models the hunting behavior of wolves, $\omega$ wolves follow $\alpha$, $\beta$ and $\delta$ wolves, which are assumed to know the location of the prey best. The equations for the hunting stage are given in Eq. (5)-(7) (Nadimi-Shahraki *et al.* 2021).

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}|, \quad \vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}|, \quad \vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \tag{5}$$

Where $\vec{D}_\alpha$, $\vec{D}_\beta$, and $\vec{D}_\delta$ are calculated using Eq. (4).

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha, \quad \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta, \quad \vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta \tag{6}$$

Here, $\vec{X}_\alpha$, $\vec{X}_\beta$, and $\vec{X}_\delta$ are the three best candidate solutions at the $t$-th iteration. $\vec{X}_1$, $\vec{X}_2$, and $\vec{X}_3$ are calculated using Eq. (6).

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{7}$$

**Attack:** The hunting process of wolves ends when the prey stops moving and then the attack process begins. This process is mathematically expressed as the decrease of the $\vec{a}$ value from 2 to 0 throughout the iterations.

**Initialization phase:** In this phase, $N$ wolves are randomly created in the search space as shown in Eq. (8) (Nadimi-Shahraki *et al.* 2021).

$$X_{i,j} = lb_j + rand \times (ub_j - lb_j) \tag{8}$$

Where $dim$ represents the dimension of the problem. The fitness value of $X_i$ is evaluated according to the fitness function representing the location of the $i$-th wolf in the $t$-th iteration.

**Movement phase:** In addition to the group hunting strategy, the dimensional learning-based hunting (DLH) strategy has been added for individual hunting in this phase. In the DLH phase, each wolf is learned by neighboring wolves as another candidate for the new current location of $X_i$ (Nadimi-Shahraki *et al.* 2021).

**DLH search strategy:** This strategy, which is used to prevent the diversity in the population from decreasing, also takes into account other wolves in the population when updating the location. In the DLH search strategy, the location of the wolf $X_{i-DLH}$ is calculated using Eq. (11). A randomly selected wolf from the population and different neighbors are used when updating the location. In this strategy, another candidate solution called $X_{i-DLH}$ is produced for the wolf $X_i$ in addition to $X_{i-GWO}$. First, the value of $R_i(t)$, which is the Euclidean distance between the candidate solution $X_i(t)$ and $X_{i-GWO}(t)$, is calculated using Eq. (9) (Nadimi-Shahraki *et al.* 2021).

$$R_i(t) = ||X_i(t) - X_{i-GWO}(t)|| \tag{9}$$

Then, the neighbors of $X_i(t)$ indicated by $N_i(t)$ are found using Eq. (10) depending on the radius of $R_i(t)$.

$$N_i(t) = \{X_j(t) \mid D_i(X_i(t), X_j(t)) \leq R_i(t), \quad X_j(t) \in Pop\} \tag{10}$$

Here, $D_i$ represents the distance between $X_i$ and $X_j$. After the neighborhood $N_i(t)$ is created, multi-neighbor learning is calculated using Eq. (11). The $d$-th dimension of $X_{i-DLH}$ is calculated using the neighbor $X_{n,d}(t)$ randomly selected from $N_i(t)$ and the randomly selected $X_{r,d}(t)$ from the population.

$$X_{i-DLH,d}(t+1) = X_{i,d}(t) + rand \times (X_{n,d}(t) - X_{r,d}(t)) \tag{11}$$

**Selection and update phase:** This is the phase where the most suitable individual is determined according to the fitness values of two candidates, $X_{i-GWO}$ and $X_{i-DLH}$.

$$X_i(t+1) = \begin{cases} X_{i-GWO}(t+1) & \text{if } f(X_{i-GWO}) < f(X_{i-DLH}) \\ X_{i-DLH}(t+1) & \text{otherwise} \end{cases} \tag{12}$$

Finally, if the fitness value of the obtained $X_{new}$ is better than $X_i$, the value of $X_i$ is updated as the new location. Otherwise, $X_i$ remains unchanged. After all these operations are applied to each candidate solution in the population, the iteration number is increased by one and the process repeats until the termination conditions are met (Nadimi-Shahraki *et al.* 2021). The flow diagram of IGWO is shown in Figure 2.

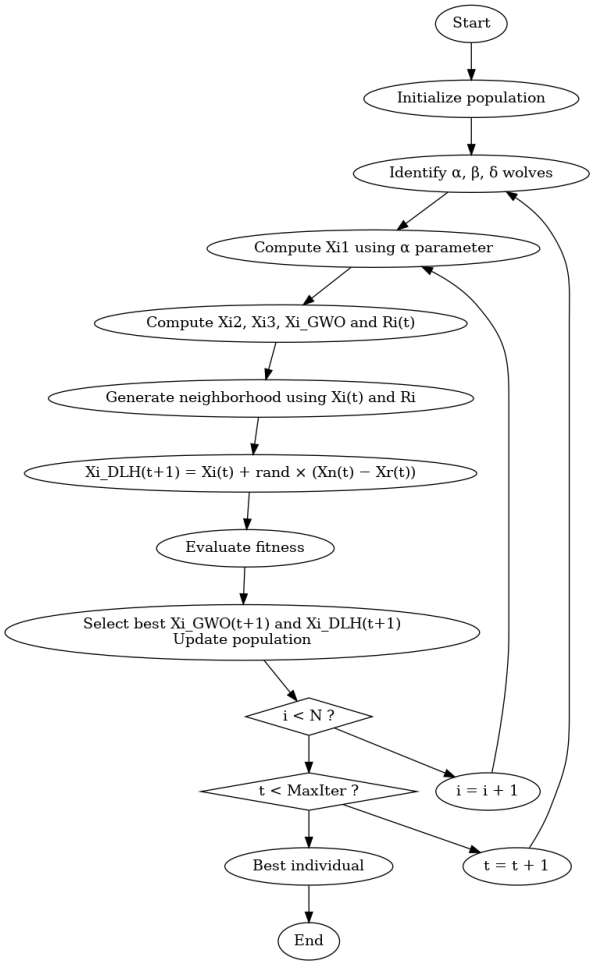



**Figure 2** Flow chart of the IGWO algorithm

The $a$, $c$ and $X_0$ values in the formula used by the LCG model constitute the decision variables of IGWO. The $n$ value is taken as a constant of 10. 30 individuals are produced as the initial population and the termination condition of the algorithm is determined as 100 iterations. The aim of this study is to ensure that when 10000 numbers are generated, each number comes at an equal rate. Since the $n$ value is taken as 10, the aim is to find the $a$, $c$ and $X_0$ values that will ensure that each of the numbers from 0 to 9 comes approximately 1000 times. For this purpose, Eq. (13) is used as the fitness function.

$$Minimize\ F = \sum_{k=0}^{n-1} (Count_k - 1000)^2 \tag{13}$$

The optimization results obtained using IGWO are analyzed based on the best and average fitness values over 50 independent runs. The results are shown in Table 1. In each column of the table, the values of $a$, $c$ and $X_0$ are given respectively. In all runs, all numbers between 0-9 were obtained 1000 times.

## CONCLUSION

In this study, the problem of selecting suitable parameters for Linear Congruential Generators was formulated as a heuristic optimization task. The Improved Grey Wolf Optimizer was employed to search for optimal multiplier, increment, and seed values. Unlike approaches based on a single evaluation criterion, the proposed framework evaluates generator quality using multiple statistical performance metrics in addition to the objective fitness function.

**Table 1** 50 $a, c$ and $X_0$ values that gave the optimum result

| $a, c, X_0$ | $a, c, X_0$ | $a, c, X_0$ | $a, c, X_0$ | $a, c, X_0$ |
|---|---|---|---|---|
| 6, 1, 5 | 1, 3, 5 | 7, 7, 5 | 7, 7, 9 | 1, 9, 3 |
| 5, 5, 7 | 4, 7, 3 | 1, 7, 4 | 1, 1, 2 | 1, 9, 4 |
| 1, 3, 6 | 1, 1, 5 | 1, 9, 10 | 8, 1, 10 | 6, 9, 2 |
| 1, 3, 2 | 7, 7, 3 | 8, 1, 3 | 1, 1, 4 | 6, 6, 7 |
| 6, 6, 1 | 5, 7, 0 | 1, 7, 5 | 3, 7, 5 | 4, 1, 1 |
| 2, 5, 4 | 5, 7, 0 | 1, 1, 9 | 1, 9, 3 | 1, 9, 4 |
| 1, 3, 8 | 5, 7, 4 | 1, 3, 10 | 2, 1, 6 | 1, 7, 7 |
| 1, 7, 6 | 1, 1, 3 | 5, 7, 8 | 1, 3, 2 | 1, 7, 3 |
| 5, 1, 5 | 8, 9, 6 | 5, 1, 7 | 1, 1, 1 | 1, 3, 5 |
| 1, 7, 9 | 1, 3, 8 | 5, 1, 7 | 1, 9, 2 | 1, 7, 1 |

The determined optimal LCG configurations tend to have the desired uniform distribution, low seriality level, and long period. The obtained results will increase the reliability of random number generators used in scientific calculations and simulations. LCG configurations with good statistical properties are important for obtaining more accurate and repeatable results. The experimental results obtained from multiple independent runs indicate that IGWO provides a stable and effective search mechanism for optimizing LCG parameters under the defined fitness function. The results indicate that guided heuristic optimization can provide more reliable parameter selections for LCGs.

Although the present study focuses on the optimization of LCG parameters using the Improved Grey Wolf Optimizer and evaluates performance based on a fitness-driven criterion, several directions remain open for future research. First, the proposed framework can be extended by incorporating additional heuristic optimization algorithms, such as genetic algorithms or particle swarm optimization, in order to perform a comprehensive comparative analysis. Second, the evaluation methodology may be enriched by employing multiple statistical performance metrics, including goodness-of-fit tests and serial correlation measures, to provide a more detailed assessment of randomness quality. Finally, future studies may benefit from expanding the reference framework by systematically integrating recent advances in heuristic optimization and random number generation literature. These extensions are expected to further strengthen the robustness and generalizability of the proposed approach.

### Acknowledgments

This study was supported by Fırat University Scientific Research Projects Coordination Unit (FUBAP) with project number MF.24.52, TEKF.24.27 and TÜBİTAK with project number 123R055.

### Ethical standard

The authors have no relevant financial or non-financial interests to disclose.

### Availability of data and material

Not applicable.

### Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## LITERATURE CITED


Akyol, S., 2022 A new hybrid method based on aquila optimizer and tangent search algorithm for global optimization. Journal of Ambient Intelligence and Humanized Computing **13**: 1–21.

Akyol, S. and B. Alatas, 2017 Plant intelligence based metaheuristic optimization algorithms. Artificial Intelligence Review **47**: 417–462.

Almaraz Luengo, E. and J. Román Villaizán, 2023 Cryptographically Secured Pseudo-Random Number Generators: Analysis and Testing with NIST Statistical Test Suite. Mathematics **11**: 4812.

Bikos, A., P. E. Nastou, G. Petroudis, and Y. C. Stamatiou, 2023 Random Number Generators: Principles and Applications. Cryptography **7**: 54.

Chen, S. and J. Zheng, 2024 A hybrid grey wolf optimizer for engineering design problems. Journal of Combinatorial Optimization **47**: 86.

Crocetti, L., S. Di Matteo, P. Nannipieri, L. Fanucci, and S. Saponara, 2022 Design and Test of an Integrated Random Number Generator with All-Digital Entropy Source. Entropy **24**: 139.

Ferreira, M. J., N. A. Silva, A. N. Pinto, and N. J. Muga, 2023 Statistical Validation of a Physical Prime Random Number Generator Based on Quantum Noise. Applied Sciences **13**: 12619.

Foreman, C., R. Yeung, and F. J. Curchod, 2024 Statistical Testing of Random Number Generators and Their Improvement Using Randomness Extraction. Entropy **26**: 1053.

Hajduk, Z., 2024 Field-Programmable Gate Array-Based True Random Number Generator Using Capacitive Oscillators. Electronics **13**: 4819.

Irfan, M., A. Ali, M. A. Khan, M. Ehatisham-ul Haq, S. N. Mehmood Shah, *et al.*, 2020 Pseudorandom Number Generator (PRNG) Design Using Hyper-Chaotic Modified Robust Logistic Map (HC-MRLM). Electronics **9**: 104.

L'Ecuyer, P., 1999 Tables of linear congruential generators of different sizes and good lattice structure. Mathematics of Computation **68**: 249–260.

Li, S., Z. Lin, Y. Yang, and R. Ning, 2024 A High-Performance FPGA PRNG Based on Multiple Deep-Dynamic Transformations. Entropy **26**: 671.

Liu, J., Z. Liang, Y. Luo, L. Cao, S. Zhang, *et al.*, 2021 A Hardware Pseudo-Random Number Generator Using Stochastic Computing and Logistic Map. Micromachines **12**: 31.

Maksymovych, V., M. Shabatura, O. Harasymchuk, R. Shevchuk, P. Sawicki, *et al.*, 2022 Combined Pseudo-Random Sequence Generator for Cybersecurity. Sensors **22**: 9700.

Mirjalili, S., S. M. Mirjalili, and A. Lewis, 2014 Grey wolf optimizer. Advances in Engineering Software **69**: 46–61.

Nadimi-Shahraki, M. H., S. Taghian, and S. Mirjalili, 2021 An improved grey wolf optimizer for solving engineering problems. Expert Systems with Applications **166**: 113917.

Nam, J.-W., J. Kim, and J.-P. Hong, 2022 Stochastic Cell- and Bit-Discard Technique to Improve Randomness of a TRNG. Electronics **11**: 1735.

Nannipieri, P., S. Di Matteo, L. Baldanzi, L. Crocetti, J. Belli, *et al.*, 2021 True Random Number Generator Based on Fibonacci-Galois Ring Oscillators for FPGA. Applied Sciences **11**: 3330.

Ozkaynak, F., 2014 Cryptographically secure random number generator with chaotic additional input. Nonlinear Dynamics **78**: 2015–2020.

Ozkaynak, F., 2015 A novel method to improve the performance of chaos based evolutionary algorithms. Optik **126**: 5434–5438.

Ozkaynak, F., 2020 A Novel Random Number Generator Based on Fractional Order Chaotic Chua System. Elektronika ir Elektrotechnika **26**: 52–57.

Ozkaynak, F. and S. Yavuz, 2013 Security problems for a pseudorandom sequence generator based on the Chen chaotic system. Computer Physics Communications **184**: 2178–2181.

Panda, A. K. and K. C. Ray, 2020 A coupled variable input LCG method and its VLSI architecture for pseudorandom bit generation. IEEE Transactions on Instrumentation and Measurement **69**: 1011–1019.

Park, S., K. Kim, K. Kim, and C. Nam, 2022 Dynamical Pseudo-Random Number Generator Using Reinforcement Learning. Applied Sciences **12**: 3377.

Qiu, Y., X. Yang, and S. Chen, 2024 An improved gray wolf optimization algorithm solving to functional optimization and engineering design problems. Scientific Reports **14**: 14190.

Rajwar, K., K. Deep, and S. Das, 2023 An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges. Artificial Intelligence Review **56**: 13187–13257.

Rojas-Muñoz, L. F., S. Sánchez-Solano, M. C. Martínez-Rodríguez, and P. Brox, 2022 True Random Number Generation Capability of a Ring Oscillator PUF for Reconfigurable Devices. Electronics **11**: 4028.

Steele Jr., G. L. and S. Vigna, 2022 Computationally easy, spectrally good multipliers for congruential pseudorandom number generators. Software: Practice and Experience **52**: 443–458.

Yu, M., J. Xu, W. Liang, *et al.*, 2024 Improved multi-strategy adaptive Grey Wolf Optimization for practical engineering applications and high-dimensional problem solving. Artificial Intelligence Review **57**: 277.

Zhao, W., Z. Chang, C. Ma, and Z. Shen, 2023 A Pseudorandom Number Generator Based on the Chaotic Map and Quantum Random Walks. Entropy **25**: 166.

Zhao, W. and C. Ma, 2024 Modification of Intertwining Logistic Map and a Novel Pseudo Random Number Generator. Symmetry **16**: 169.

Álvarez, R., F. Martínez, and A. Zamora, 2022 Improving the Statistical Qualities of Pseudo Random Number Generators. Symmetry **14**: 269.



***How to cite this article:*** Akyol, S., and Özkaynak, F. Determination of Optimal Linear Congruence Generators Parameters with Heuristic Optimization Algorithms. *ADBA Computer Science*, 3(1), 1-5, 2026.




ADBA Computer Science
Cutting-Edge Scientific Solutions

e-ISSN: 3023-8544
RESEARCH ARTICLE
Vol.3 / No.1 / 2026 / pp.6-12
https://doi.org/10.69882/adba.cs.2026012

# Hyperparameter Optimization for Big Data: Adapting Sampling Methods to Apache Spark MLlib

**M. Maruf Öztürk** *,[1]
*Department of Computer Engineering, Engineering and Natural Sciences Faculty, Suleyman Demirel University, Isparta, Türkiye.

**ABSTRACT** MLlib is an Apache Spark library that provides many machine learning algorithms and data processing utilities. Although the default configuration of these algorithms yields satisfactory results for practitioners, further tuning is often needed to improve resource usage efficiency. Furthermore, tuned MLlib algorithms may run faster than those using default configurations. However, this improvement depends on several factors, including machine settings, dataset design, and operating system preferences. Previous studies have generally focused on developing sophisticated tuners for MLlib, evaluating algorithm-focused optimizers for their competitiveness. Although derivative-based and model-free optimizers have been modified for use with MLlib, sampling-based optimizers are generally overlooked. To fill this research gap, this study empirically compares sampling-based and model-free techniques for tuning MLlib. Firstly, Monte Carlo and Cross-Entropy sampling algorithms are adapted to optimize MLlib algorithms. Subsequently, model-free techniques, including grid and random search algorithms, are compared with these sampling-based algorithms. Through extensive experimentation, their advantages and limitations are highlighted. Finally, threats to validity and future directions for unlocking the tuning potential of Apache Spark are discussed by interpreting performance bottlenecks and promising areas for optimization.

**KEYWORDS**
Optimization
Apache spark
Tuning
Monte carlo
Cross-entropy

## INTRODUCTION

Hyperparameter optimization (HO) is an indispensable yet intricate process in machine learning (ML) (Feurer and Hutter 2019). For this reason, several HO techniques exist, including random (Khaldi et al. 2025) and grid search (Ilmeboya et al. 2024) algorithms, Bayesian optimization (Eleftheriadis et al. 2024), the Nelder-Mead method (Herbst et al. 2024), and other heuristics (van Stein et al. 2024). Model-free HO techniques, such as grid search, are commonly embedded into automated ML (AutoML) systems. This integration facilitates the entire HO process by enabling a discussion of the advantages and disadvantages of existing alternatives.

Apache Spark is a big data processing framework with over 200 configurable hyperparameters (Meng et al. 2016). To achieve maximum efficiency, the default configuration of these hyperparameters should be compared with optimized ones. However, selecting the right optimization algorithm is a complex and effort-intensive process. This complexity arises because the Resilient Distributed Dataset (RDD), a core component of Apache Spark, requires a significant amount of time to solve even a moderate optimization problem. To address this issue, preliminary experiments are often performed to predict overall performance. Moreover, the parallel execution of hyperparameter searches can shorten the optimization process (Meister et al. 2020).

MLlib was originally designed for performing machine learning tasks, including feature selection, classification, regression, and clustering (Assefi et al. 2017). Although it yields satisfactory results with the default hyperparameters proposed by Apache Spark, resource management issues such as CPU usage, execution time, and memory consumption can often only be controlled by configuring MLlib's hyperparameters. For model-free tuning, random search may be a good solution when the tuning budget is limited. Although such techniques have achieved significant reductions in execution time, their performance depends greatly on the sampling conducted within the local search space (Nguyen et al. 2018). The number of possible configurations is very large, so any comprehensive search technique may require several days to find an optimal set of parameters. Reducing this search space may be possible by using a Bayesian algorithm to eliminate redundant candidate points (Cao et al. 2024).

Cross-entropy (CE) is a sampling-based minimization algorithm consisting of two main phases: generating random instances and updating model parameters (Benham et al. 2017). It has been applied to various optimization problems, including manufacturing (Beruvides et al. 2016), sensor management (Naeem et al. 2009), engineering design (Li et al. 2019), and path planning (Tang et al. 2024). CE is very effective in landscapes with many local optima, such as those found when optimizing Apache Spark.

Monte Carlo is a sampling-based optimization algorithm that predicts possible outcomes by repeating simulations. It leverages probability distributions to produce outputs. In particular, the Hamiltonian version of Monte Carlo is suitable for complex, real-world optimization problems (Campbell et al. 2021). Note that the scale factor can significantly improve the general performance of the Monte Carlo method.

**Manuscript received:** 12 December 2025,
**Revised:** 13 January 2026,
**Accepted:** 13 January 2026.

[1]muhammedozturk@sdu.edu.tr (**Corresponding author**)

Although each HO technique has distinctive characteristics that can be observed during optimization (Andonie 2019), big data processing platforms require sophisticated libraries to perform tuning. For instance, Apache Spark MLlib only supports cross-validation and the generation of a parameter grid for hyperparameters (Zhu et al. 2025). Therefore, the tuning process becomes computationally expensive due to data-based parallelism (Zhou et al. 2025). Consequently, there is a need to perform HO with techniques compatible with RDDs. This approach can remarkably reduce the time allocated for HO. Alternatively, custom-tuned models can be used to replace the default algorithms provided by MLlib.

Despite the common use of model-free HO techniques, it remains unclear whether sampling-based methods, such as Monte Carlo and cross-entropy, are superior. To bridge this gap, we perform a comprehensive comparison of model-free and sampling-based algorithms. To this end, four classification datasets are processed using MLlib algorithms alongside hyperparameter sets proposed by the comparison algorithms. The convergence, execution time, and rejection rate of four optimization algorithms Monte Carlo, grid search, random search, and CE were evaluated.

The paper claims the contributions as follows: 1) We investigate Apache MLlib whether it is compatible with simulation-based optimization, 2) We compare Monte Carlo with traditional techniques to reveal the most suitable hyperparameter optimization technique for Apache Spark, 3) We present a comprehensive complexity analysis of the baselines for the practitioners. The remainder of the paper is organized as follows: Section 2 gives definitions of the background. Section 3 summarizes related work. Section 4 elucidates the proposed method. The results are presented in Section 5. Threats to validity are discussed in Section 6. Finally, Section 7 concludes the paper and summarizes the results.

## BACKGROUND

**Monte Carlo expectation:** Monte Carlo runs for approximating a random process in a specific number of iteration. A good estimation can be calculated as $\theta_n = \sum_{i=1}^{n} P_i/n$ in which $n$ is the number of points and $P$ is a candidate point.

**Cross-entropy minimizer:** Let $S$ be a set of states and $f$ is the performance function used in $S$. The main aim is to minimize $f$ over $S$ in which a corresponding minimizer $c*$ is responsible for the success of $f$. ($f(c*) = min_{f(X)}, X \in S$)

**Cross-entropy convergence**: Convergence of cross-entropy can be calculated with a chain rule that utilizes some partial derivations: $\frac{\partial L}{\partial w} = (-y(1-p) + (1-y)p) \cdot x$ $\frac{\partial L}{\partial w} = (-y + yp + p - yp) \cdot x$ $\frac{\partial L}{\partial w} = (p-y) \cdot x$ $w_{new} = w_{old} - \eta \cdot (p-y) \cdot x$ $b_{new} = b_{old} - \eta \cdot (p-y)$ (since $\frac{\partial z}{\partial b} = 1$)

in which $y$ and $p$ denote the actual and predicted label of prediction, respectively. $w$ is the weight used in model update that is strongly related to model bias $b$.

**Importance sampling estimator:** A local optima $\gamma$ determines the importance of sampling $\frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{g(X_i)} . f(X_i < \gamma)$ in which $g$ represents sampling density.

**Hyperparameter optimizer:** Let $\sigma(x)$ is an algorithm that produces an error $E$. The hyperparameter optimizer $H_{opt}$ aims to minimize $E$ ($H_{opt} \rightarrow min_{\sigma(x)}$) in which $x$ is a set of hyperparameters. Some specific number of subsets $U_1, U_2, ..., U_m$ may constitute $x$.

## LITERATURE REVIEW

Random search was devised to overcome the disadvantages of grid and manual search in large parameter spaces (Bergstra and Bengio 2012). It requires less computational time yet is not suitable for fully controllable experiments, as the number of iterations is set by the practitioner at the beginning. Compared to random search, Bayesian optimization has more potential to achieve sample efficiency for a large number of function evaluations (Turner et al. 2021). Since grid search is time-consuming to evaluate all solution candidates, randomizing the hyperparameter set offers a practical alternative. Hence, a threshold may halt the execution before evaluating the entire grid.

Campbell et al. (2021) developed a Monte Carlo-based hyperparameter optimization method, proposing some updates for the Adam optimizer. They found that Monte Carlo was remarkably helpful for choosing the initial distribution in real-world problems. Large and complex parameter spaces cannot be comprehensively analyzed by exhaustive search algorithms such as grid search. To solve this issue, Baziar et al. (2025) predicted urban water resources using machine learning algorithms, revealing the advantages of Monte Carlo over alternatives. In numerical simulation, Svensson et al. (2015) found that sequential Monte Carlo imposes less computational load compared to Hamiltonian and adaptive importance sampling. In real-world optimization problems, the difference between Monte Carlo and AUTO-SKLEARN Feurer et al. (2022) becomes visible at later stages of tuning (Rakotoarison et al. 2019). Furthermore, although Monte Carlo has the potential to be a good alternative to Bayesian models, it also poses a risk of overfitting. In Jalobeanu et al. (2002), a regularization method was proposed for restoring noisy images. To this end, Monte Carlo was leveraged to predict sampling density. The main limitation of that method is the need for prior knowledge when a unique solution space is desired. When the outputs of a model are very noisy, employing Monte Carlo to predict possible outcomes is a wise choice (Dunbar et al. 2025). However, large-scale hyperparameter transfer methods may be required to know the prior distribution.

Cross-entropy method was originally devised as a minimization algorithm proposed by Rubinstein (1997). It was then used for density estimation tasks in classification problems (Kurian et al. 2021). Cross-entropy is also a feasible algorithm for multi-objective optimization. It yielded promising results for robot path planning (Tang et al. 2024), where various physical constraints were considered during optimization. However, it remains unclear whether the search efficiency depends on the recombination probability of certain adjustment factors. In image processing, cross-entropy strongly depends on the square root error, which leads to an unfavorable convergence rate (Mao et al. 2023). To solve this problem, a generalized bound may be defined for loss minimizers. Cross-entropy may be combined with other probability estimation methods, such as Monte Carlo (Liu et al. 2021). The computational burden stemming from high-dimensional integrals can thus be alleviated. However, its reliability was only tested on numerical functions rather than real-world problems.

## METHOD

An overview of the experiment is provided in Figure 1. The process consists of three main steps: data preparation, tuning of MLlib, and results production. It is worth noting that these processes are cyclical, allowing for the creation of a dataset-based configuration pathway to replicate the experiment. RDD conversion is necessary to utilize MLlib algorithms, as they are designed to operate on a

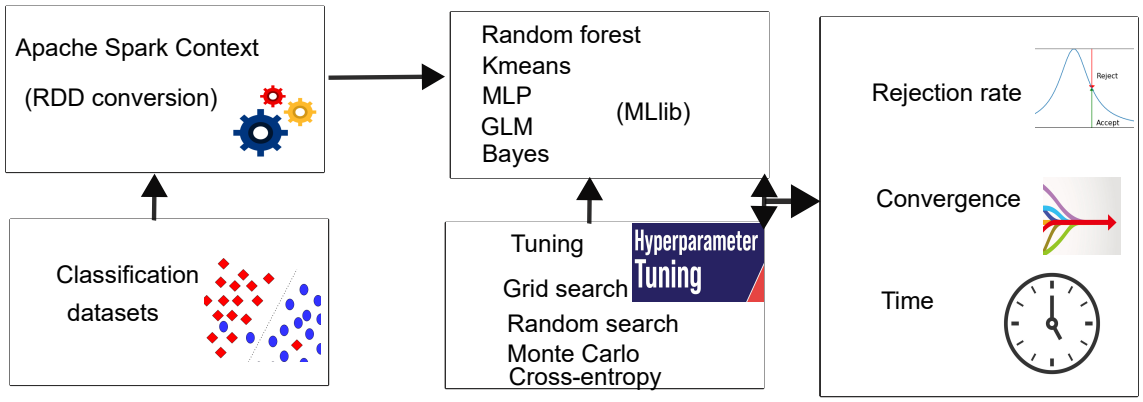



**Figure 1** Overview of the experiment.

worker-based structure. Tuning is limited to certain model-based algorithms and two sampling methods. To evaluate performance comprehensively, the experiment was not restricted to a single type of dataset, such as classification. Consequently, the prediction performance of the algorithms was not included in the experimental design.

Algorithm 1 details the use of Monte Carlo methods for optimizing MLlib hyperparameters. Here, *nrep* represents the number of Monte Carlo repetitions; the computational burden of optimization can therefore be adjusted by modifying this parameter. Step 3 establishes a connection to the Apache Spark context. Step 4 converts the dataset into a resilient distributed dataset (RDD). Step 5 splits the dataset into training and testing sets to perform cross-validation. The *result* variable contains the optimal hyperparameters and their corresponding rejection rates.

**Algorithm 1** Spark-based MLlib Optimization with Monte Carlo Simulation

1: **Input: nrep, parameters, MLlib algorithm (sparkm), data set (D)**
2: **Output: optimal settings, rejection rates**
3: sc ← spark_connect(master = "local")
4: D_rdd ← copy_to(sc, D)
5: train, test ← sdf_random_split(D_rdd
6: result ← MonteCarlo(sparkm, train, test, parameters)
7: return(result)

In Algorithm 2, *rho* is the elite proportion that denotes the lowest cost of population. *iterThr* is the termination threshold for the optimization iteration. *CEoptim* is a function that is available in R *CEoptim* library. Steps 3-5 are the same with those of Algorithm 1. In Step 7, convergence and ultimate hyperparameter set are returned. R versions of two proposed algorithms can be accessed through [1].

### Datasets

Table 1 provides details of the experimental datasets.

The *Dense* dataset was originally created to improve neural network models for classification[2]. Its 30 numerical features can be used to predict a label, making it suitable for binary classification. The *Microsoft* dataset was extracted from a security contest and

[1] https://github.com/muhammedozturk/samplingBasedOptimization
[2] https://www.kaggle.com/c/dense-network/data?select=train.csv

**Algorithm 2** Spark-based MLlib Optimization with CE

1: **Input: rho, N, iterThr, parameters, MLlib algorithm (sparkm), data set (D)**
2: **Output: convergence, optimum of function**
3: sc ← spark_connect(master = "local")
4: D_rdd ← copy_to(sc, D)
5: train, test ← sdf_random_split(D_rdd)
6: result ← CEoptim(sparkm, train, test, parameters)
7: return(result)

is publicly available[3]. With 1,804 numerical features (187.83 MB), it can be exploited for various classification experiments. The *Payload* dataset was presented by Politecnico di Milano University and created for a project[4]. Its 31 numerical features can be utilized for classification tasks where data churn is very low (+10, -1). The *Santander* dataset was generated from user transactions[5]. Its 140 numerical features, which consist of floating-point numbers, can be used to design machine learning experiments.

**Table 1** MLlib data sets utilized in this work.

| Name | Instances | Type |
|---|---|---|
| Dense | 175000 | Classification |
| Microsoft | 10868 | Classification |
| Payload | 130529 | Classification |
| Santander | 200000 | Classification |

### Experimental settings

Four MLlib algorithms are evaluated in the experiment: Bayes, Kmeans, multi layer perceptron (MLP), and generalized linear model (GLM). Since this study does not consider initialization of

[3] https://www.kaggle.com/muhammad4hmed/malwaremicrosoftbig
[4] https://zenodo.org/record/5731597
[5] https://www.kaggle.com/datasets/lakshmi25npathi/santander-customer-transaction-prediction-dataset

an optimizer, the tuners are run in default configurations. Otherwise, it is another research direction to find the optimal execution settings Monte Carlo and cross-entropy.

Table 2 presents the details of hyperparameters tuned by the comparison algorithms. Note that Bayes has only one hyperparameter (smoothing) that hardens to generalize the result. On the other hand, some hyperparameters such as $max_iter$ and $tol$ are common for GLM, MLP, Kmeans.

**Table 2** Details of the hyperparameters used in the experiment.

| Algorithm | Hyperparameter | Step size | Range |
|---|---|---|---|
| Random forest | num_trees | 1 | 10-50 |
| Random forest | max_depth | 1 | 2-10 |
| Random forest | subsampling_rate | 0.01 | 0.1-0.9 |
| Bayes | smoothing | 1 | 1-5 |
| GLM | max_iter | 1 | 10-100 |
| GLM | tol | 1e-1 | 1e-09:1e-01 |
| MLP | max_iter | 1 | 10-50 |
| MLP | step_size | 0.01 | 0.1-0.7 |
| MLP | tol | 1e-1 | 1e-09:1e-01 |
| Kmeans | tol | 1e-1 | 1e-09:1e-01 |
| Kmeans | max_iter | 1 | 10-50 |
| Kmeans | k | 1 | 1-5 |

The machine utilized to perform the experiment has following properties: CentOS Linux, 64-bit, Intel(R) Xenon(R) 2.9 GHz, 24 CPU Cores server with 222 GB RAM. The results given in next section refers to the average values obtained from four datasets.

## RESULTS

Figure 2 shows the execution time of the cross-entropy algorithm for 10,000 training instances. Note that each method quickly reaches its peak point before the 200th call. Random Forest requires the most time compared to the others. K-means is the first method that reaches a stagnant line, which occurs at the 140th call. Broadly speaking, it can be concluded that unsupervised methods require less time for cross-entropy optimization. Furthermore, MLP and Bayes have similar fluctuations and cross paths at the 800th call. They can be used interchangeably in Spark for hyperparameter optimization.

The convergence analysis shown in Figure 3 indicates that an rapid drop occurs within a few iterations, regardless of the method used. Optimization time is highly dependent on the algorithm type. In stark contrast, the convergence rate is not affected by the number of iterations and follows nearly the same path throughout the optimization process. It is worth noting that various parameters impact this convergence behavior. First, the experimental data consists primarily of binary classification datasets. Second, MLlib does not allow for the execution of every machine learning algorithm. Instead, it provides a set of well-known algorithms, many of which date back to the early days of machine learning. Consequently, the experiment should be extended to include various deep learning architectures; however, it is currently limited to a feedforward artificial neural network.

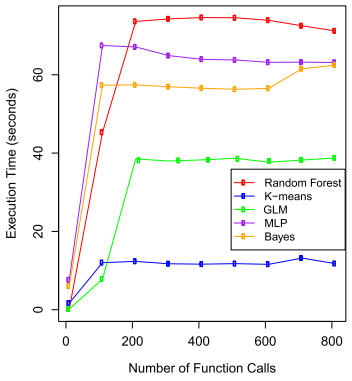



**Figure 2** Execution time analysis of MLlib algorithms.

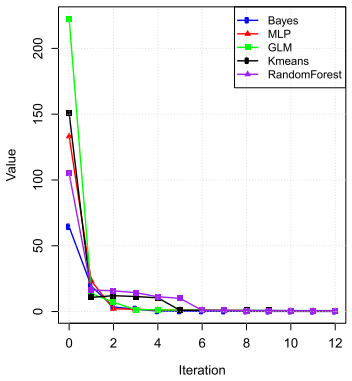



**Figure 3** Convergence of MLlib algorithms for cross-entropy optimization.

Figure 4 shows the time competitiveness of the optimizers. Random search is a very stable algorithm, thanks to its threshold settings for iterations. Despite the fact that sampling-based methods provide a critical opportunity to delve into the rationale behind optimal settings beyond mere performance measures, they take much more time than model-free algorithms. In this respect, sampling-based methods should be taken into consideration only when there are restricted computational resources. If an in-depth analysis is not needed to evaluate the hyperparameters, grid search is a good alternative to alleviate the computational burden.

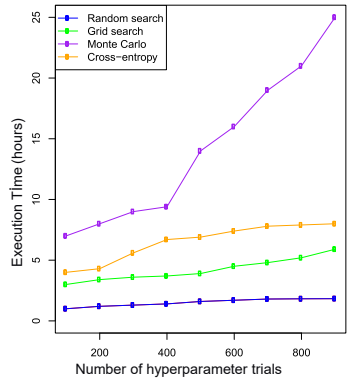



**Figure 4** Time comparison of tuners.

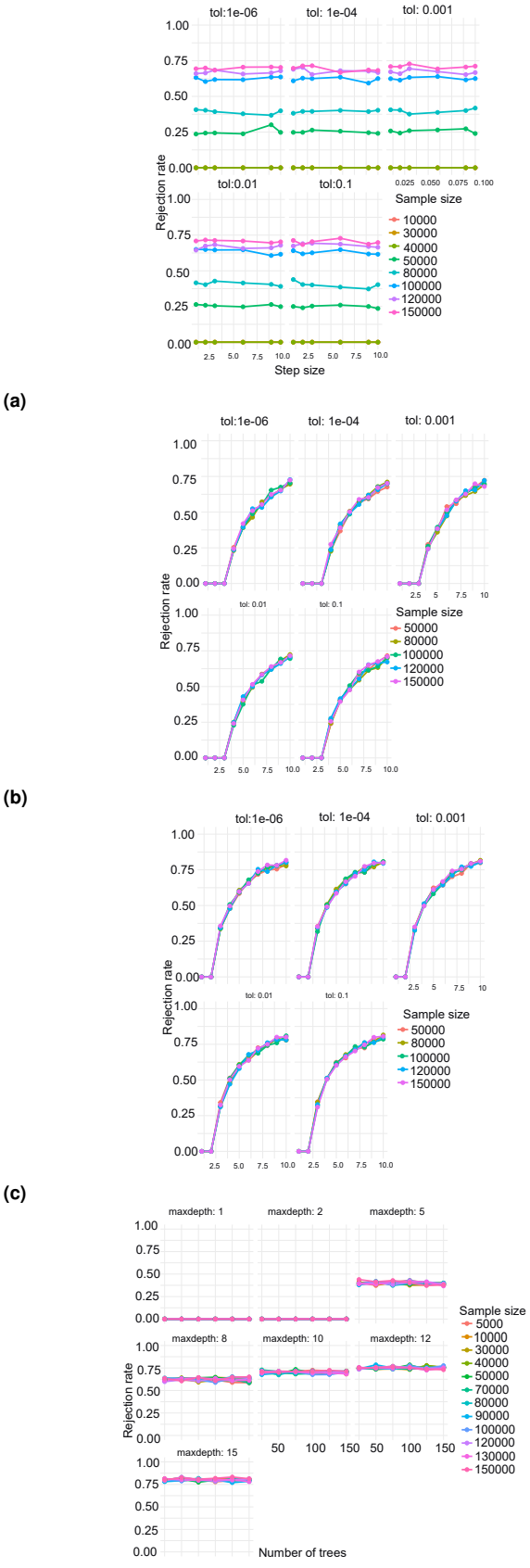



**Figure 5** a) MLP b) Kmeans c) GLM d) Random forest

Figure 5 demonstrates the rejection rates of the comparison algorithms for different sample sizes. It is worth noting that a low rejection rate does not necessarily result in a reliable set of optimal hyperparameters. For MLP, a sample size of 10,000 is sufficient to find optimal configurations. The step size is not a critical factor affecting the Monte Carlo process, and the case is similar for the tolerance (tol) parameter. The reasonable number of clusters for K-Means is seven; however, the sample size has a negligible effect on the Monte Carlo performance for this algorithm. A similar pattern was detected for GLM, where 7.5 is the optimal regularization parameter and the number of instances is also not particularly important. For Random Forest, the maximum tree depth should be kept at five to avoid reducing the rejection rate. One can conclude that complex algorithms, such as MLP, are more sensitive to the sampling rate compared to traditional machine learning algorithms in Apache Spark.

### Quantitative Analysis of Optimization Challenges in Spark

Table 3 shows the average time per iteration for each optimizer. Monte Carlo and Grid search methods exhibit a 1% and 5% higher average time per iteration compared to Random Search, quantifying the computational overhead introduced by their sampling mechanisms within Spark's RDD-based architecture. On the other hand, Cross-entropy is the second preferable method when considering time-related computational burden. Analysis of Spark event logs revealed that tasks for the Monte Carlo method spent approximately 15% of their runtime on deserialization and data preparation, compared to 5% for Grid Search, indicating the overhead introduced by frequent parameter distribution in sampling-based approaches. While the previous sections analyzed the intrinsic properties of the optimizers, practitioners often operate under strict computational constraints. To provide actionable guidance, we conducted a budget-aware comparison. We imposed three realistic wall-clock time budgets—30, 60, and 120 minutes—on the tuning process for each optimizer and MLlib algorithm. For each run, we recorded the best validation score achieved before the budget was exhausted. The aggregated results are presented in Figure 6.

**Table 3** Time analysis based on per iteration.

| Algorithm | Avg. Time per Iteration (minutes) | Std. Dev. (minutes) |
|---|---|---|
| Monte Carlo | 1.4 | 0.4 |
| Grid search | 0.9 | 0.3 |
| Random search | 0.4 | 0.1 |
| Cross-entropy | 0.6 | 0.2 |

## DISCUSSION

Time complexity is a good indicator for choosing a hyperparameter optimization (HO) method. In this respect, four methods were evaluated based on their time complexities. Grid search has a complexity of $O(N * D)$, where $N$ is the number of candidate values per hyperparameter and $D$ denotes the number of dimensions (hyperparameters). For this reason, grid search is very sensitive to the size of the dimension. The time complexity of random search can be represented as $O(T)$, where $T$ is the number of trials, which limits the total execution time. Monte Carlo has the same complexity, $O(T)$, where $T$ denotes the number of samplings. Note, however,

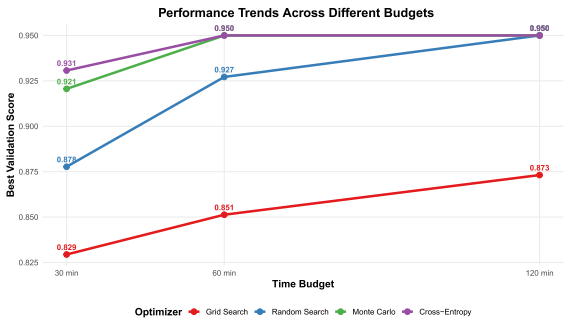



**Figure 6** Budget-aware comparison between the optimization techniques .

that Monte Carlo required much more time than random search for tuning due to compatibility issues in Apache Spark. To generalize the results, future experiments could be replicated without using a third-party library to execute Monte Carlo. The highest time complexity belongs to cross-entropy method, at $O(T * M * K)$, where $T$ is the number of iterations, and $K$ and $M$ denote the number of best and elite samples, respectively. If a pre-defined region is used for the optimization, cross-entropy can find an optimal solution at a low cost. Otherwise, it is not very effective for searching the entire solution space.

There are some threats to the validity: 1) The experiment is limited to classification datasets; future work could extend it to other data types, such as regression or clustering datasets. 2) The tuning performance of MLlib was evaluated with various techniques. However, MLlib's native library does not support all machine learning algorithm types. To overcome this issue, its scope should be extended beyond the currently supported feedforward neural network to include other modern architectures.

Our experimental results validate this claim. As shown in Figure 6 and Table 3, Random Search (O(T)) consistently provided the best increase for validation score under the tightest time budgets, making it the most practical choice when computational resources are limited. Conversely, while the Cross-Entropy method has a higher per-iteration cost (O(TMK)), its superior sample efficiency allowed it to achieve the best overall performance when a more generous budget was available. This demonstrates how understanding time complexity guides the practitioner's trade-off between immediate results and ultimate solution quality.

## CONCLUSION

Hyperparameter Optimization (HO) is a technique used to enhance the performance of machine learning models. While various HO techniques have been proposed to address specific engineering problems, it remains ambiguous whether sampling-based methods outperform model-free methods. To address this question, this paper presents a comparative analysis between sampling-based and model-free HO approaches. According to the obtained results: 1. Generalized Linear Models (GLM) exhibit fast convergence when combined with cross-entropy optimization. 2. K-means is highly resilient to an increasing number of function calls. 3. Although Monte Carlo sampling provides valuable insights into hyperparameter interactions through its rejection rates, it requires substantial time to complete sampling for each target hyperparameter. The work presented here can be extended through several future research avenues: 1. Derivative-based optimization methods could be compared with sampling-based approaches to develop new HO strategies. 2. The cost-effectiveness of HO could be evaluated on serverless computing platforms. 3. While MLlib currently provides cross-validation for model optimization, Apache Spark should be enriched with native, data-parallel HO libraries instead of relying on third-party solutions to accelerate the tuning process.

### Ethical standard

The author has no relevant financial or non-financial interests to disclose.

### Availability of data and material

Not applicable.

### Conflicts of interest

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

## LITERATURE CITED


Andonie, R. (2019). Hyperparameter optimization in learning systems. *Journal of Membrane Computing*, 1(4):279–291.

Assefi, M., Behravesh, E., Liu, G., and Tappert, A. R. (2017). Big data machine learning using apache spark mllib. In *Proceedings of the IEEE International Conference on Big Data*, pages 3492–3498.

Baziar, M., Kashi, A. R., and Karimi, S. M. H. (2025). Machine learning-based monte carlo hyperparameter optimization for thms prediction in urban water distribution networks. *Journal of Water Process Engineering*, 73:107683.

Benham, T., Cole, J. S., and Kloeden, P. E. (2017). Ceoptim: Cross-entropy r package for optimization. *Journal of Statistical Software*, 76:1–29.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(1):281–305.

Beruvides, G., Quiza, R., and Castaño, F. (2016). Multi-objective optimization based on an improved cross-entropy method: A case study of a micro-scale manufacturing process. *Information Sciences*, 334:161–173.

Campbell, A., Chen, J., and Li, W. (2021). A gradient based strategy for hamiltonian monte carlo hyperparameter optimization. In *Proceedings of the International Conference on Machine Learning*, pages 1238–1248.

Cao, R., Chen, Y., and Zhang, W. (2024). Etune: Efficient configuration tuning for big-data software systems via configuration space reduction. *Journal of Systems and Software*, 209:111936.

Dunbar, O., Garcia-Trillos, N., and Perego, M. (2025). Hyperparameter optimization for randomized algorithms: a case study on random features. *Statistics and Computing*, 35(3):1–28.

Eleftheriadis, P., Lioudakis, G., and Amditis, A. (2024). Joint state of charge and state of health estimation using bidirectional lstm and bayesian hyperparameter optimization. *IEEE Access*, 12:80244–80254.

Feurer, M., Eggensperger, K., Falkner, S., Lindauer, M., and Hutter, F. (2022). Auto-sklearn 2.0: Hands-free automl via meta-learning. *Journal of Machine Learning Research*, 23(261):1–61.

Feurer, M. and Hutter, F. (2019). Hyperparameter optimization. In *Automated Machine Learning: Methods, Systems, Challenges*, pages 3–33. Springer, Cham, Switzerland.

Herbst, P., Schuld, M., and Petruccione, F. (2024). On optimizing hyperparameters for quantum neural networks. In *Proceedings of the IEEE International Conference on Quantum Computing and Engineering (QCE)*, volume 1, pages 1478–1489.

Ilmeboya, J., Adebayo, P. O., and Adewumi, A. O. (2024). Hyperparameter tuning in machine learning: A comprehensive review. *Journal of Engineering Research and Reports*, 26(6):388–395.

Jalobeanu, A., Zerubia, J., and Blanc-Talon, J. (2002). Hyperparameter estimation for satellite image restoration using a mcmc maximum-likelihood method. *Pattern Recognition*, 35(2):341–352.

Khaldi, M., Kafi, A. E., and Bernoussi, S. E. (2025). Hyperparameter optimization for malicious url detection: Leveraging optuna and random search in machine learning and deep learning models. *Informatica*, 49(27).

Kurian, N. C., Dubey, S. R., and Chakraborty, S. (2021). Sample specific generalized cross entropy for robust histology image classification. In *Proceedings of the IEEE 18th International Symposium on Biomedical Imaging*, pages 1934–1938.

Li, G., Zhao, W., and Wang, Y. (2019). An improved butterfly optimization algorithm for engineering design problems using the cross-entropy method. *Symmetry*, 11(8):1049.

Liu, X., Zhang, Y., and Chen, W. (2021). A cross-entropy algorithm based on quasi-monte carlo estimation and its application in hull form optimization. *International Journal of Naval Architecture and Ocean Engineering*, 13:115–125.

Mao, A., Mohri, M., and Abu-Mostafa, Y. S. (2023). Cross-entropy loss functions: Theoretical analysis and applications. In *Proceedings of the International Conference on Machine Learning*, pages 23803–23828.

Meister, M., Parnell, T., and Zhang, C. (2020). Maggy: Scalable asynchronous parallel hyperparameter search. In *Proceedings of the 1st Workshop on Distributed Machine Learning*, pages 28–33.

Meng, X., Bradley, J. K., Yavuz, B., Sparks, E. R., Venkataraman, S., and Franklin, M. J. (2016). Mllib: Machine learning in apache spark. *Journal of Machine Learning Research*, 17(34):1–7.

Naeem, M., Reza, S. M. S., and Kemp, A. H. (2009). Cross-entropy optimization for sensor selection problems. In *Proceedings of the 9th International Symposium on Communications and Information Technology*, pages 396–401.

Nguyen, N., Hassan, M. A., Bai, K., and Wang, Y. (2018). Towards automatic tuning of apache spark configuration. In *Proceedings of the IEEE 11th International Conference on Cloud Computing (CLOUD)*, pages 417–425.

Rakotoarison, H., Sebag, M., and Schoenauer, M. (2019). Automated machine learning with monte-carlo tree search. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 3296–3303.

Rubinstein, R. Y. (1997). Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 99(1):89–112.

Svensson, A., Schön, T. B., and Kok, M. (2015). Marginalizing gaussian process hyperparameters using sequential monte carlo. In *Proceedings of the IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 477–480.

Tang, Q., Wang, Y., and Liu, Z. (2024). A dual-robot cooperative arc welding path planning algorithm based on multi-objective cross-entropy optimization. *Robotics and Computer-Integrated Manufacturing*, 89:102760.

Turner, R., Eriksson, D., McCourt, M., Kiili, J., Laaksonen, E., and Xu, Z. (2021). Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. In *NeurIPS 2020 Competition Demonstration Track*, pages 3–26.

van Stein, N., Bäck, T., and Emmerich, M. (2024). In-the-loop hyper-parameter optimization for llm-based automated design of heuristics. *ACM Transactions on Evolutionary Learning*.

Zhou, M., Li, Y., and Chen, W. (2025). Towards hybrid architectures for big data analytics: Insights from spark-mpi integration. *IEEE Transactions on Services Computing*, pages 1852–1868.

Zhu, Y., Wang, X., and Zhang, L. (2025). Rockhopper: A robust optimizer for spark configuration tuning in production environment. In *Companion Proceedings of the International Conference on Management of Data*, pages 743–756.

***How to cite this article:*** Öztürk, M. M. Hyperparameter Optimization for Big Data: Adapting Sampling Methods to Apache Spark MLlib. *ADBA Computer Science*, 3(1), 6-12, 2026.



ADBA Computer Science
Cutting-Edge Scientific Solutions

e-ISSN: 3023-8544
RESEARCH ARTICLE
Vol.3 / No.1 / 2026 / pp.13-17
https://doi.org/10.69882/adba.cs.2026013

# Microcontroller Execution, Random Number Generation and Chaos Annihilation in Piecewise Quadratic Map

**Lucienne Makouo** *,1, **Jules Metsebo** α,2, **Cyrille Ainamon** β,3, **Oumate Alhadji Abba** §,4 **and André Chéagé Chamgoué** **,5

*Department of Civil Engineering and Architecture, National Higher Polytechnic Institute, University of Bamenda, P.O. Box 39, Bamenda, Cameroon, αDepartment of Hydraulics and Water Management, National Advanced School of Engineering, University of Maroua, P.O Box 46 Maroua, Cameroon, βInstitut de Mathématiques et de Sciences Physiques, Université d'Abomey-Calavi, BP 613, Porto Novo, Benin, §Department of Physics, Faculty of Science, University of Maroua, P.O. Box 814, Maroua, Cameroon, **Department of Petroleum and Gas Engineering, School of Geology and Mining Engineering, University of Ngaoundere, P.O. BOX 115, Meiganga, Cameroon.

**ABSTRACT** This paper presents the study of microcontroller execution, pseudo random number generator (PRNG) and chaos annihilation in a piecewise quadratic map (PQM). PQM is generated by replacing the cubic nonlinearity of cubic map by absolute nonlinearity. Bistable outphase and monostable chaotic characteristics and bistable outphase periodic oscillations are encountered in PQM during numerical simulations. The microcontroller execution of PQM is realized to validate the numerical results encountered in PQM. The PRNG derived from the PQM is made and the NIST 800-22 statistical test validated it. Finally, analytical calculations and numerical simulations show the effectiveness of chaos annihilation in PQM using feedback controller.

**KEYWORDS**
Chaotic maps
Microcontroller execution
RNG
Chaos annihilation

## INTRODUCTION

Discrete chaotic maps are usually iterated mathematical function that display chaotic behavior. They have aroused the interest of physicists, engineers, mathematicians, chemists, population biologists, and many others (Bergé *et al.* 1987; Moon 1987; Zeng *et al.* 1990). They have a common characteristic, namely nonlinearity and their behavior depends on certain controllable quantities. Nonlinearity is a characteristic of a system or function where the output is not a simple multiple of the input (Abdullaev *et al.* 2002). The nonlinearity can be classified as smooth and nonsmooth functions (Rapcsák 2008; Aguirre 2014). Piecewise nonlinearities belong to a class of nonsmooth functions used to describe dynamical systems having intermittent contact, as mechanical systems with moving parts (Zhang *et al.* 2022; Chávez *et al.* 2016).

The most popular chaotic map is the logistic map which first appeared during the study of population in ecology (May 1987). It is nowadays used to describe many natural processes and has been the subject of numerous studies (Chen *et al.* 2021; Das *et al.* 2010; Dua *et al.* 2024). Logistic chaotic map has smooth function as nonlinearity which is quadratic function. Cubic map belongs to the same universality class as the logistics map (Ambika and Joseph 1992). Cubic chaotic map has smooth function as nonlinearity which is cubic function. Since the cubic map has been proposed in (Rogers and Whitley 1983; Zeng 1985) which demonstrated the existence of a period doubling route to chaos. It has been subjected to detailed studies (Steeb *et al.* 1997; Wang *et al.* 2021). A piecewise cubic map (PCM) described by piecewise-nonlinear function with two parameters has been proposed in (Udwadia and Guttalu 1989). The authors of (Udwadia and Guttalu 1989) investigated analytically and numerically the PCM to find the route to chaos. The motivation for this research paper stems from the Lozi chaotic map (Lozi 2023) where Lozi replaced the quadratic term in the Henon chaotic map (García-Martínez and Campos-Cantón 2015) by an absolute value function.

Since 1978, Lozi chaotic map has been extremely studied and continues to be. Hundreds of publications have investigated its particular characteristics and applied its properties in many fields via algorithms (Lozi 2023). Inspired by Lozi chaotic map, this paper designs a PQM generated by replacing the cubic nonlinearity of cubic map by absolute nonlinearity. It studies the microcontroller execution, PRNG and chaos annihilation in designed PQM. The tri-modal map derived from the logistic map which belongs to the family of chaotic PQM has been proposed in (García-Martínez and Campos-Cantón 2015). The authors of (García-Martínez and Campos-Cantón 2015) has been demonstrated that it was pos-

**Manuscript received:** 4 November 2025,
**Revised:** 16 December 2025,
**Accepted:** 30 December 2025.

[1]makouol@yahoo.fr (**Corresponding author**).
[2]jmetsebo@gmail.com
[3]ainamoncyrille@yahoo.fr
[4]oumat_oaa@yahoo.fr
[5]acchamgoue@gmail.com

sible to have PRNG based on chaotic tri-modal map. The one-dimensional piecewise map derived from the logistic map has been proposed in (Cassal-Quiroga *et al.* 2022).

The authors of (Cassal-Quiroga *et al.* 2022) has been demonstrated analytically and numerically that the proposed piecewise map was able to display bistability phenomenon, coexisting and chaotic attractors. This research paper demonstrated: (1) numerically and experimentally bistable outphase and monostable chaotic characteristics and bistable outphase periodic oscillations in PQM and; (2) the possibility to have PRNG based on chaotic PQM and (3) the chaos annihilation in PQM using a feedback controller. This paper is organized in the following manner: The second section deals with the microcontroller execution of the designed PQM. The PRNG based on designed PQM is studied in the third section. The fourth section deals with the chaos annihilation in the designed PQM. The fifth section concludes this paper.

## MICROCONTROLLER VALIDATION OF PQM

The PQM is designed by replacing the term $x_n^3$ in the cubic map (Wang *et al.* 2021) by the term $x_n\,|x_n|$. It is described by:

$$x_{n+1} = \beta x_n - x_n\,|x_n| \tag{1}$$

where $x_n$ and $x_{n+1}$ respectively represent the $n^{th}$ and $(n+1)^{th}$ states with $n$ being a natural number and $\beta$ is positive constant. The PQM described by map $(1)$ can be also written as follows:

$$x_{n+1} = \begin{cases} \beta x_n + x_n^2, & x_n < 0, \\ \beta x_n - x_n^2, & x_n \geq 0. \end{cases} \tag{2}$$

The dynamical characteristics encountered in PQM are depicted in Fig. 1.

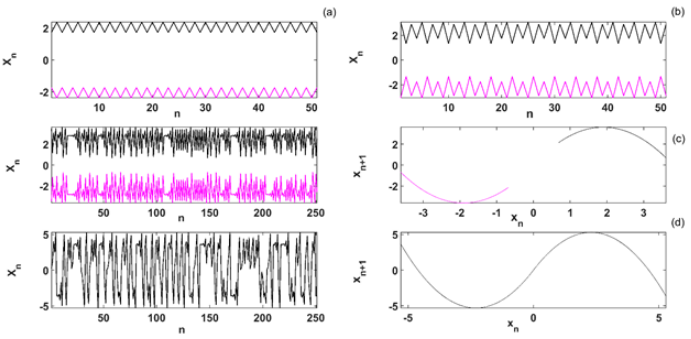



**Figure 1** Dynamical characteristics of PQM for specific values of $\beta$ : (a) $\beta = 3.1$ , (b) $\beta = 3.6$ , (c) $\beta = 3.8$ and (d) $\beta = 4.6$ . The plots in black are generated using the initial states: $x_0 = 0.1$ whereas the plots in magenta are generated with the initial states $x_0 = -0.1$.

The PQM displays bistable outphase period-1-oscillations in Fig. 1 (a), bistable outphase period-2-oscillations in Fig. 1 (b), bistable outphase chaotic characteristics in Fig. 1 (c) and monostable chaotic characteristics in Fig. 1 (d). The experimental set-up diagram of PQM microcontroller execution is elaborated in Figure 2.

The practical set-up of Fig. 2 is constituted of a digital platform based on the STM32F4xx series board, incorporating an ARM Cortex M4-based microcontroller for signal processing and a RIGOL DHO804 digital oscilloscope for visualization of phase portraits

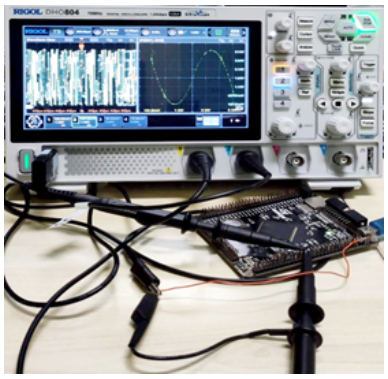


**Figure 2** Arrangement of the equipment for the microcontroller execution of PQM.

and time-series data. The Arduino code is then written, complied, and the resulting binary file by the device are converted to analog using the integrated digital-to-analog converter of the STM32F4xx board. Figs 3 and 4 illustrate time-series and function given by the PQM from the microcontroller execution, showing different qualitative behavior of PQM.

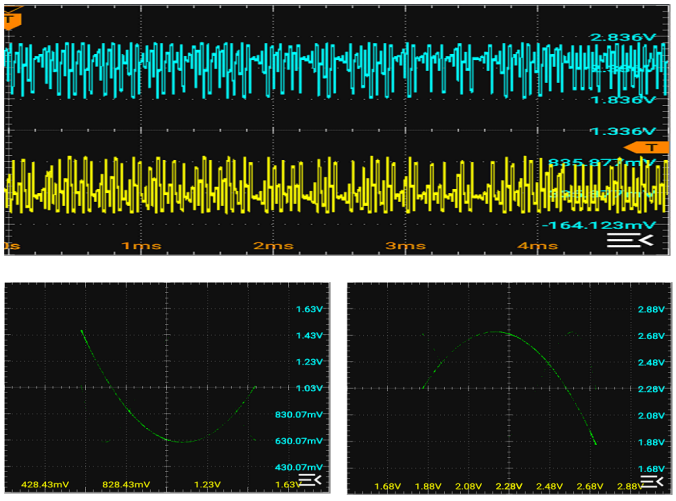



**Figure 3** Bistable outphase chaotic characteristics via the microcontroller execution of PQM.

Figures 3 and 4 display bistable and monostable chaotic characteristics, respectively gotten from the microcontroller execution of PQM described by map (1). The microcontroller results of Figs 3 and 4 validate the numerical results of Figs 1 (c) and 1 (d).

## RANDOM NUMBER GENERATOR BASED ON PQM

In this following, the design process of a PQM-based PRNGs is presented and the resulting module is tested following the widely used NIST 800-22 package. It is a battery of up to fifteen statistical tests to evaluate the random nature of a Boolean sequence for cryptography purpose. The NIST 800-22 tests analyze the properties of a long binary array by checking the presence or the absence of specific patterns. All the tests must be successful without exceptions to validate a given binary sequence as random. The computed values of the state variable xn are scaled up and converted on sixteen bits.

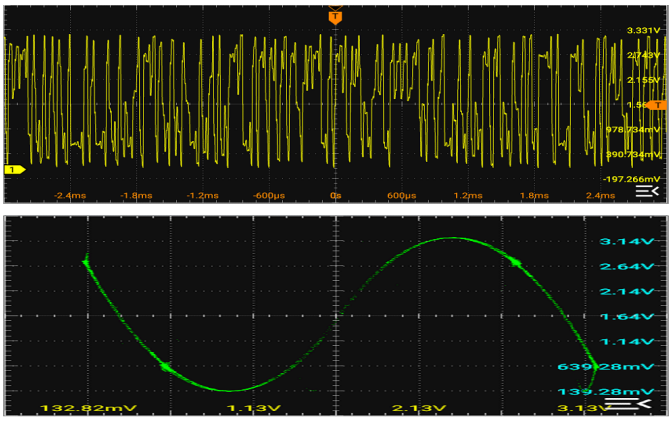


**Figure 4** Monostable chaotic characteristics via the microcontroller execution of PQM.

Subsequently, the eight least significant binary digits B are considered and re-ordered to obtain an 8-bit output key K. The process is repeated until the required data length is attained. The collected binary sequence is tested and validated with the NIST test suite. If a single test failed, scaling factor textitkp, map parameter $\beta$, initial condition $x(0)$ and the arrangement of computed output bits are fine-tuned to achieve better results. Upon successful, the generated sequences' randomness is validated, confirming the PRNG design's success. The operation of the designed PQM-PRNG is described in the pseudocode of Algorithm 1.

**Algorithm 1** Pseudocode of the Designed PQM-PRNG

Initialize parameters: $k_p = 2^{32}$, $\beta = 4.6$, $result = 1.0$
Set initial condition: $x = 0.1$
$n\_bits \leftarrow 0$
**while** $n\_bits < 1{,}000{,}000$ **do**
    $x \leftarrow \beta \cdot x - x \cdot |x|$
    $result \leftarrow k_p \cdot x$
    $X \leftarrow float\_to\_binary(result, 16)$
    $B \leftarrow lsb(X, 8)$
    $K \leftarrow reorder(B)$
    Append $K$ to file `binarysequence.bin`
    $n\_bits \leftarrow n\_bits + 8$

In Algorithm 1, the functions float_to_binary, *lsb* and reorder are used to respectively convert a real number into a 16-bit binary sequence, extract the 8 least significant bit of a 16-bit sequence and rearrange the element of an 8-bit binary number. Several option of re-ordering are tested and the following re-arrangement: $K(1) \leftarrow B(5)$; $K(2) \leftarrow B(3)$; $K(3) \leftarrow B(7)$; $K(4) \leftarrow B(1)$; $K(5) \leftarrow B(8)$; $K(6) \leftarrow B(4)$; $K(7) \leftarrow B(6)$; $K(8) \leftarrow B(2)$; ensures good reults for the chosen set of parameters. Fig. 5 illustrates the design flowchart of the PQM-based PRNG.

Among the numerous possible combination, a set of parameters allowing the designed PRNG to validate all NIST 800-22 test is: $k_p = 2^{32}$, $x_n(0) = 0.1$, $\beta = 4.6$ and the output re-arranged as $\{Key\} = \{X_1X_2X_3X_4X_5X_6X_7X_8\}$. The randomness is determined for each test by assessing the p-value, with the significance threshold set at 0.01 suggesting that the data's randomness is evaluated with a 99 confidence. A 1Mbit of data file containing the binary sequence generated from PQM-based PRNG is loaded and tested.

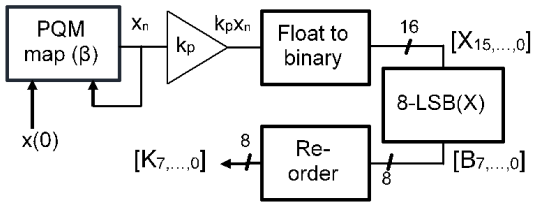



**Figure 5** Functional block diagram of the proposed PQM-based PRNG.

The NIST evaluation results are given in Table 1.

**Table 1** NIST Evaluation Results

| Test Name | P-value | Result |
|---|---|---|
| Frequency | 0.4703 | Passed |
| Block Frequency | 0.4094 | Passed |
| Runs | 0.9977 | Passed |
| Longest Run | 0.9145 | Passed |
| Matrix Rank | 0.1205 | Passed |
| Spectral | 0.7342 | Passed |
| Non-overlapping Template Matching | 0.5682 | Passed |
| Overlapping Template Matching | 0.2358 | Passed |
| Universal | 0.7965 | Passed |
| Linear Complexity | 0.0422 | Passed |
| Serial | 0.8595 | Passed |
| Approximate Entropy | 0.9048 | Passed |
| Cumulative Sums (Cusum) | 0.8110 | Passed |
| Random Excursions | 0.3607 | Passed |
| Random Excursions Variant | 0.2824 | Passed |

The $p$-values of the $x_n$ signal pass successful the fifteen statistical evaluations of NIST 800-22 as shown in Table 1. The random signal generated from chaotic PQM can be used in applications such as secure communication schemes and other chaos-based applications.

## CHAOS ANNIHILATION IN PQM

Section 4 is devoted to the context of chaos annihilation which is different to the context of generating pseudorandom numbers. A feedback controller (Wang and Wang 2008) is used to annihilate chaos in PQM in this section. The controlled PQM can be described by:

$$x_{n+1} = \beta x_n - x_n |x_n| + (\hat{x} - \kappa x_n) \tag{3}$$

where $\kappa$ is the feedback coefficient and $\hat{x}$ stands for the fixed points of map (1): $\hat{x} = 0$ and $\hat{x} = \pm(\beta - 1)$. By choosing to stabilize PQM described by the map (3) at using the feedback controller, the Jacobian matrix of map (3) at $\hat{x} = 0$ is:

$$J_{(\hat{x}=0)} = (\beta x - |x| - xsign(x) - \kappa)_{(\hat{x}=0)} = \beta - \kappa \tag{4}$$

If $|\beta - \kappa| < 1$, i.e. $\beta - 1 < \kappa < \beta + 1$ , map (3) will be stabilized at a fixed point $\hat{x} = 0$. The bifurcation diagram of map (3) and Lyapunov exponent (LE) with respect to the feedback coefficient $\kappa$ in Fig. 6.

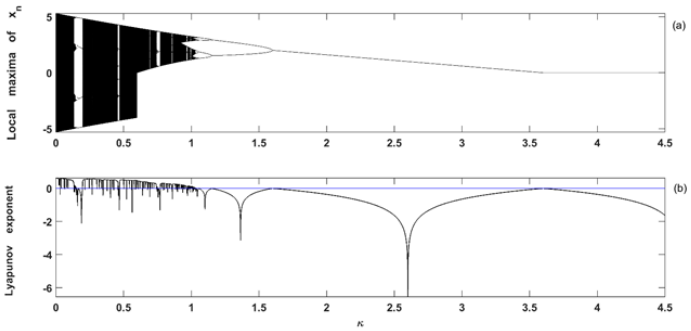



**Figure 6** Local maxima of $x_n(a)$ and LE (b) versus the feedback coefficient $\kappa$ for $\beta = 4.6$, $\hat{x} = 0$ and the initial states: $x_0 = 0.1$.

The controlled PQM described in map (3) exhibits reverse period doubling to chaotic region interspersed by periodic windows, limit cycle, Hopf bifurcation $\kappa \approx 3.6$ and fixed point as illustrated in Fig. 6 (a). The dynamical characteristics encountered in Fig. 6 (a) are corroborated by its corresponding LE in Fig. 6 (b). Fig. 6 reveals that the feedback controller can annihilate the chaos encountered in PQM. From Fig. 6 (a), one can see that for $\kappa = \beta - 1 = 3.6$ map (3) has a Hopf bifurcation, for $\kappa > \beta + 1 = 5.6$ map (3) is divergent and for $3.6 < \kappa < 5.6$ is stabilized at $\hat{x} = 0$ . Therefore, the simulations results are consistent with the analytical calculations. Furthermore, the effects of chaos annihilations in PQM using the feedback controller are more illustrated in the time series as shown in Fig. 7.

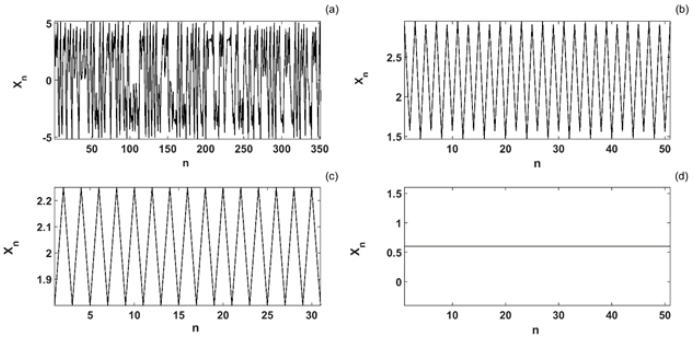



**Figure 7** Times series of $x_n$ for specific values of $\kappa$ : (a) $\kappa = .0.05$ , (b) $\kappa = 1.15$ , (c) $\kappa = 1.55$ and (d) $\kappa = 3.5$ . The others parameters are $\beta = 4.6$ , $\hat{x} = 0$ and the initial states: $x_0 = 0.1$.

The amplitude of $x_n$ exhibits chaos in Fig. 7 (a), period-2-oscillations in Fig. 7 (b), limit cycle in Fig. 7 (c) and fixed point in Fig. 7 (d).

## CONCLUSION

This paper explored the microcontroller execution, pseudo random number generator (PRNG) and chaos annihilation in a piecewise quadratic map (PQM). PQM was designed by replacing the cubic term of cubic map by absolute nonlinearity. During numerical simulations, PQM revealed the monostable and bistable outphase chaotic characteristics and bistable outphase periodic characteristics. The numerical results encountered in PQM was confirmed through microcontroller execution and the two results were found to be in perfect agreement. The randomness of the PRNG was extensively tested using the NIST 800-22 test suite, confirming the suitability of PQM-based PRNG for applications such as secure communication schemes and other chaos-based applications. It was demonstrated that the analytical calculations and numerical simulations confirmed the effectiveness of chaos annihilation in PQM using the feedback controller.

### Ethical standard

The authors have no relevant financial or non-financial interests to disclose.

### Availability of data and material

Not applicable.

### Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## LITERATURE CITED


Abdullaev, F., O. Bang, and M. P. Sørensen, 2002 *Nonlinearity and Disorder: Theory and Applications*, volume 45. Springer Science & Business Media.

Aguirre, L. A., 2014 Identification of smooth nonlinear dynamical systems with non-smooth steady-state features. Automatica **50**: 1160–1166.

Ambika, G. and K. B. Joseph, 1992 Deterministic chaos in one-dimensional maps—the period doubling and intermittency routes. Pramana **39**: 193–252.

Bergé, P., Y. Pomeau, and C. Vidal, 1987 Order within chaos new york.

Cassal-Quiroga, B., H. Gilardi-Velázquez, and E. Campos-Cantón, 2022 Multistability analysis of a piecewise map via bifurcations. International Journal of Bifurcation and Chaos **32**: 2250241.

Chávez, J. P., Y. Liu, E. Pavlovskaia, and M. Wiercigroch, 2016 Path-following analysis of the dynamical response of a piecewise-linear capsule system. Communications in Nonlinear Science and Numerical Simulation **37**: 102–114.

Chen, S., S. Feng, W. Fu, and Y. Zhang, 2021 Logistic map: Stability and entrance to chaos. In *Journal of Physics: Conference Series*, volume 2014, p. 012009, IOP Publishing.

Das, N., R. Sharmah, and N. Dutta, 2010 Period doubling bifurcation and associated universal properties in the verhulst population model. International J. of Math. Sci. & Engg. Appls **4**: 1–14.

Dua, M., R. Bhogal, S. Dua, and N. Chakravarty, 2024 Satellite image encryption using amalgamation of randomized three chaotic maps and dna encoding. Physica Scripta **100**: 015241.

García-Martínez, M. and E. Campos-Cantón, 2015 Pseudo-random bit generator based on multi-modal maps. Nonlinear Dynamics **82**: 2119–2131.

Lozi, R., 2023 Survey of recent applications of the chaotic lozi map. Algorithms **16**: 491–553.
May, R. M., 1987 Chaos and the dynamics of biological populations. Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences **413**: 27–44.
Moon, F. C., 1987 Chaotic vibrations: An introduction for applied scientists and engineers new york.
Rapcsák, T., 2008 Smooth nonlinear nonconvex optimization. In *Encyclopedia of Optimization*, pp. 3622–3625, Springer.
Rogers, T. D. and D. C. Whitley, 1983 Chaos in the cubic mapping. Mathematical Modelling **4**: 9–25.
Steeb, W., F. Solms, T. K. Shi, and R. Stoop, 1997 Cubic map, complexity and ljapunov exponent. Physica Scripta **55**: 520.
Udwadia, F. E. and R. S. Guttalu, 1989 Chaotic dynamics of a piecewise cubic map. Physical Review A **40**: 4032.
Wang, C., Y. Di, J. Tang, J. Shuai, Y. Zhang, *et al.*, 2021 The dynamic analysis of a novel reconfigurable cubic chaotic map and its application in finite field. Symmetry **13**: 1420.
Wang, X. and M. Wang, 2008 Chaotic control of logistic map. Modern Physics Letters B **22**: 1941–1949.
Zeng, W.-z., 1985 A recursion formula for the number of stable orbits in the cubic map. Chin. Phys. Lett. **2**: 429–431.
Zeng, X., R. Pielke, and R. Eykholt, 1990 Chaos in daisyworld. Tellus B **42**: 309–318.
Zhang, Z., J. Páez Chávez, J. Sieber, and Y. Liu, 2022 Controlling grazing-induced multistability in a piecewise-smooth impacting system via the time-delayed feedback control. Nonlinear Dynamics **107**: 1595–1610.

***How to cite this article:*** Makouo, L., Metsebo, J., Ainamon, C., Abba, O. A., and Chamgoué, A. C. Microcontroller Execution, Random Number Generation and Chaos Annihilation in Piecewise Quadratic Map. *ADBA Computer Science*, 3(1), 13-17, 2026.



ADBA Computer Science
Cutting-Edge Scientific Solutions

e-ISSN: 3023-8544
RESEARCH ARTICLE
Vol.3 / No.1 / 2026 / pp.18-25
https://doi.org/10.69882/adba.cs.2026014

# Chaos Analysis of a Fuel-Slosh Coupled Aircraft System

**Ons Kooli** *,[1] **and İhsan Pehlivan** *,[2]
*Sakarya University of Applied Sciences, Faculty of Technology, Deparment of Electrical and Electronics Engineering, Sakarya, Türkiye.

**ABSTRACT** This study investigates nonlinear and chaotic dynamics induced by fuel sloshing in a partially filled aircraft tank. The system is modeled as a rigid aircraft pitch degree of freedom coupled with a nonlinear slosh pendulum, representing the dominant liquid motion. Numerical simulations are used to analyze the system response through time series and phase portraits, revealing complex dynamical behavior. Chaotic dynamics are quantitatively confirmed using Lyapunov exponents and their parametric variation. In addition, bifurcation analyses are performed with respect to key system parameters, illustrating transitions from periodic to chaotic motion. The results highlight the significant impact of fuel slosh on aircraft pitch dynamics and emphasize the importance of accounting for nonlinear fluid–structure interactions in stability analysis and control-oriented modeling.

**KEYWORDS**
Nonlinear dynamics
Fuel sloshing
Aircraft pitch dynamics
Chaotic motion
Dynamical analysis

## INTRODUCTION

Understanding nonlinear interactions in aircraft systems is crucial for both control performance and flight safety. One important source of nonlinear behavior is fuel slosh, where the motion of liquid inside partially filled tanks couples with the rigid-body dynamics of the aircraft. This coupling can introduce additional forces and moments that significantly alter the system response, potentially leading to large oscillations, loss of stability, or even chaotic motion under relatively small excitations. Recent studies have emphasized that such nonlinearities are not merely secondary effects but can be primary drivers of complex instability in modern aerospace structures (Mahmoudvand *et al.* 2025).

Fuel sloshing dynamics are particularly relevant for large aircraft, launch vehicles, and unmanned aerial vehicles with significant fuel volumes. The pendulum-like motion of liquid with a free surface can amplify pitching, rolling, or yawing oscillations and modify the effective inertia and damping of the vehicle. Extensive research has investigated the hydrodynamic aspects of sloshing and its interaction with structural dynamics, highlighting its influence on stability, loads, and energy dissipation mechanisms (Faltinsen 2005; Ibrahim 2005). Experimental and numerical studies in aerospace applications have shown that fuel slosh can alter dynamic responses and aeroelastic characteristics depending on fill level, excitation frequency, and tank geometry (Constantin *et al.* 2022; Langlois and Kabamba 2015; Wang *et al.* 2021). Recent advancements in bifurcation analysis have further revealed that even subtle changes in flight parameters can trigger abrupt transitions in coupled fluid-structure systems (Liu and Wang 2024).

To model sloshing effects efficiently, many works have employed reduced-order mechanical analogs, such as pendulum or mass–spring representations, to approximate liquid motion without resorting to full computational fluid dynamics simulations. These simplified models have been successfully applied in spacecraft attitude dynamics and aircraft fuel tank studies, providing valuable insight into slosh-induced forces and moments while maintaining analytical and computational tractability (Schlee and Smith 2006; Savella and Ibrahim 2005; Reyhanoglu and van der Schaft 2011). Current research continues to utilize these analogs to develop robust control strategies capable of suppressing nonlinear oscillations in real-time (Chen and Zhang 2025).

Despite these advances, most existing studies focus on linearized behavior, stability margins, or damping characteristics. Comparatively fewer works explicitly address the nonlinear and chaotic dynamics that may arise from the coupling between rigid-body motion and fuel slosh under periodic excitation. Phenomena such as sensitivity to initial conditions, aperiodic oscillations, and transitions to chaos remain insufficiently explored in coupled pitch–slosh systems.

**Manuscript received:** 30 October 2025,
**Revised:** 23 December 2025,
**Accepted:** 11 January 2026.

[1]onskooli@zohomail.eu (**Corresponding author**).
[2]ipehlivan@subu.edu.tr

Motivated by this gap, the present work investigates a nonlinear aircraft pitch model coupled with a fuel slosh pendulum. The system dynamics are analyzed through equilibrium analysis, Jacobian and eigenvalue evaluation, time-series responses, phase portraits, Lyapunov exponents and spectra, as well as bifurcation diagrams. Parameter variations are explored to study their influence on system stability and chaotic behavior. The results demonstrate that fuel slosh can act as a source of strong nonlinearity capable of inducing complex and chaotic responses, emphasizing the importance of accounting for such effects in aircraft dynamic analysis, stability assessment, and control design.

## DESCRIPTION OF THE COUPLED SYSTEM

In this study, the aircraft is modeled as a rigid body undergoing pitch motion and coupled with a simplified fuel slosh representation. The liquid motion inside the fuel tank is approximated using a mechanical analog in the form of a pendulum, which captures the dominant sloshing mode associated with the free-surface motion of the fuel. This reduced-order approach allows the essential nonlinear coupling between the aircraft pitch dynamics and the internal fuel motion to be analyzed without resorting to full fluid dynamic simulations.

The coupling between the aircraft and the sloshing fuel introduces additional forces and moments that act on the pitching motion, modifying the effective inertia, damping, and stiffness of the system. Conversely, the aircraft pitch acceleration influences the slosh dynamics, resulting in a bidirectional interaction. Such mutual coupling is a known source of nonlinear behavior and may lead to complex dynamic responses, including quasiperiodic and chaotic oscillations under certain excitation conditions.

The resulting mathematical model consists of a set of nonlinear ordinary differential equations governing the aircraft pitch angle and rate, together with the angular displacement and velocity of the slosh pendulum. These equations form the basis for the analytical and numerical investigations presented in the subsequent sections.

While the single-pendulum mechanical analog effectively captures the dominant longitudinal sloshing mode and the resulting chaotic transitions, it is acknowledged that more complex approximations can provide higher-order fidelity. Advanced modeling techniques, such as multi-pendulum arrays, hydraulic mechanical models, or high-fidelity Computational Fluid Dynamics (CFD) simulations, can capture additional slosh modes and secondary nonlinearities arising from fluid viscosity and complex tank geometries. However, for the purpose of investigating the fundamental mechanisms of chaos and bifurcations in pitch-slosh coupling, the reduced-order pendulum model provides a computationally efficient and analytically tractable framework that preserves the essential nonlinearities of the system.

### Aircraft Control and Pitch Dynamics

To understand the forced pitch–actuator system analyzed in this study, it is important to review the primary control surfaces and their influence on aircraft motion. Figure 1 illustrates a schematic of a generic fixed-wing aircraft, showing the elevator, rudder, and ailerons.

Pitch and Elevator, Pitch refers to the rotation of the aircraft about its lateral axis (nose up or nose down). The *elevator*, located on the horizontal tail, is the primary control surface that commands this motion. Moving the elevator up causes the nose to rise (positive pitch), while moving it down causes the nose to drop (negative pitch).

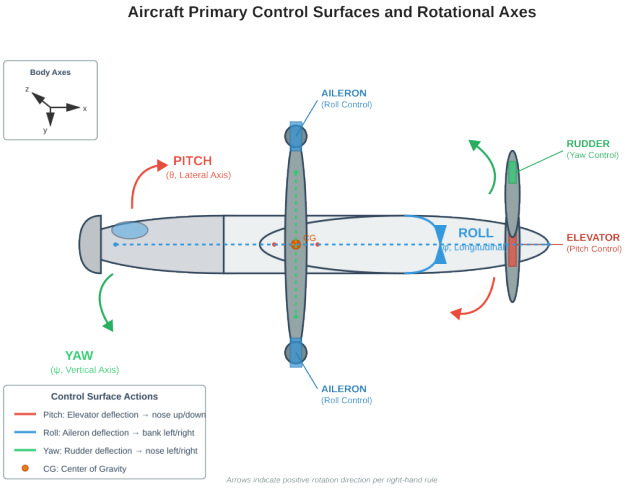



**Figure 1** Schematic of a fixed-wing aircraft showing primary control surfaces and rotational axes: pitch (elevator), roll (ailerons), and yaw (rudder).

Roll and Ailerons, Roll is the rotation about the longitudinal axis of the aircraft. The *ailerons*, located on the trailing edge of the wings, control roll. Deflecting one aileron up and the other down causes the aircraft to tilt sideways, enabling banking turns.

Yaw and Rudder, Yaw is the rotation about the vertical axis, controlling the left-right direction of the nose. The *rudder*, mounted on the vertical tail fin, produces yawing moments. Moving the rudder left or right rotates the aircraft nose in the corresponding direction.

In this work, the forced nonlinear system models the coupling between the aircraft pitch and the elevator dynamics under periodic forcing. The pitch stiffness is characterized by a cubic nonlinearity, modeled as a Duffing-type oscillator. Understanding these basic control surfaces and their associated motions is crucial for interpreting the system's response, including quasi-periodic and chaotic behaviors.

The system consists of:

- A rigid aircraft with pitch degree of freedom $\theta$, modeled with a Duffing-type cubic nonlinear stiffness.
- A partially filled fuel tank, modeled as a pendulum mass $m_s$ of length $l$ and slosh damping $c_s$.
- Nonlinear coupling between aircraft pitch acceleration and slosh motion via coupling coefficient $\beta$.
- Optional external periodic forcing representing aerodynamic gusts or engine vibrations.

### Physical Explanation

The slosh mass acts like a pendulum whose motion is affected by the aircraft pitch $\theta$ and, in turn, applies a nonlinear torque to the aircraft. This coupling introduces a strong source of nonlinearity through the interaction of the state variables and parameters. Specifically, the aircraft's longitudinal motion is defined by the pitch angle $\theta$ and angular rate $q = \dot{\theta}$, while the internal liquid dynamics are represented by the slosh pendulum angle $\phi$ and its rate $\psi = \dot{\phi}$.

The system's complexity arises from the nonlinear coupling where the aircraft acceleration $\ddot{\theta}$ influences the slosh motion via the coupling coefficient $\beta$, and the resulting slosh displacement feeds back as a corrective or perturbing torque on the pitch dynamics

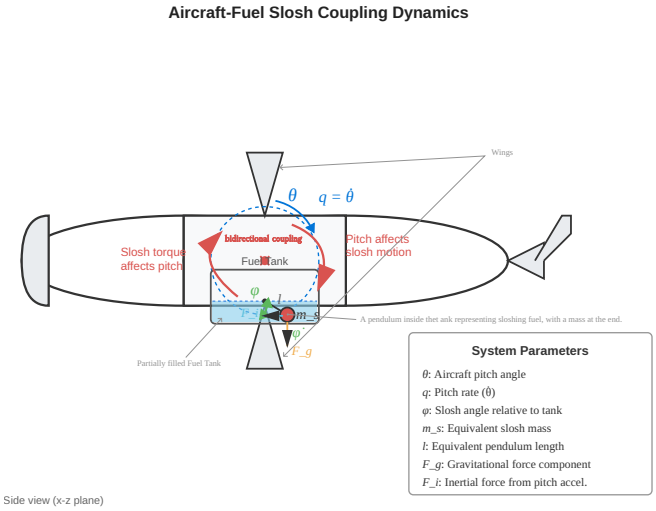



**Figure 2** Schematic of aircraft coupled with a slosh pendulum representing fuel motion.

through the coefficient $K_c$. Furthermore, the energy dissipation in the system is governed by the aircraft's aerodynamic damping ratio $\zeta$ and the slosh-specific damping coefficient $c_s$; however, while these terms reduce total energy, they do not eliminate the potential for complex aperiodic behavior or the onset of chaos.

### Fuel Tank Dynamics

The aircraft considered in this study contains a partially filled fuel tank, which introduces additional nonlinear effects due to fluid motion. The tank is typically located near the center of mass of the aircraft to minimize static stability issues; however, the dynamic motion of the liquid can generate significant forces that affect the airframe's response. For modeling purposes, the tank geometry is assumed to be rigid, possessing a rectangular or cylindrical shape. The fuel inside this volume is allowed to move freely, generating sloshing waves in response to both the aircraft's linear acceleration and rotational pitch maneuvers. In this study, the slosh degrees of freedom are primarily restricted to the longitudinal (nose–tail) direction, as this provides the most direct and significant coupling with the aircraft's pitch dynamics, whereas lateral slosh is neglected to maintain a focused analysis on longitudinal stability.

### Coupling Between Aircraft Pitch and Fuel Slosh

The motion of the fuel inside the tank interacts with the aircraft dynamics, introducing nonlinear coupling between the pitch–actuator system and the fluid motion. When the aircraft undergoes a pitch maneuver, the free surface of the fuel tilts, generating additional forces and moments that act upon the aircraft's center of mass. These slosh-induced forces subsequently modify the pitch trajectory and can excite unwanted oscillations throughout the airframe. Under specific conditions of periodic external forcing, the recursive interaction between the aircraft pitch and the internal fuel slosh can lead to highly complex dynamics, including quasi-periodic transitions or fully developed chaotic behavior.

## FUEL SLOSH MODEL AND GOVERNING DYNAMICS

The coupled aircraft pitch and fuel-slosh dynamics are modeled as a nonlinear four-dimensional system. The state variables are the

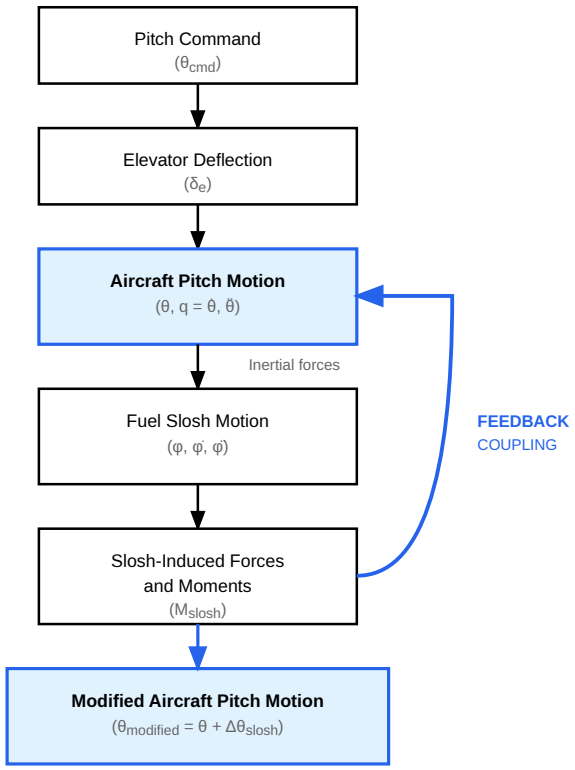



**Figure 3** Computational framework and feedback coupling logic between the aircraft pitch control and the nonlinear fuel slosh dynamics.

aircraft pitch angle $\theta$, pitch rate $q$, slosh displacement $\phi$, and slosh velocity $\psi$.

The pitch dynamics are modeled using a Duffing-type nonlinear oscillator with external harmonic excitation, a formulation commonly adopted in vibration and chaos studies of mechanical and aeroelastic systems (Beltrán-Carbajal and Silva-Navarro 2009).

$$\begin{aligned} \dot{\theta} &= q, \\ \dot{q} &= -\omega_n^2\theta + \alpha\theta^3 - 2\zeta\omega_n q + K_c\phi + F\sin(\omega t), \\ \dot{\phi} &= \psi, \\ \dot{\psi} &= -\omega_s^2\phi - c_s\psi - \beta\dot{q}, \end{aligned} \tag{1}$$

where $\omega_s$ is the natural frequency of the slosh pendulum and $c_s$ is the slosh damping coefficient. In the numerical implementation, the expression for $\dot{q}$ is substituted into the $\dot{\psi}$ equation to decouple the accelerations and form an explicit state-space representation. This substitution reveals the direct influence of the nonlinear pitch stiffness and aerodynamic damping on the fuel slosh acceleration.

The numerical simulations are conducted using a set of dimensionless parameters that represent a realistic physical configuration of a mid-sized transport aircraft. The mass ratio between the sloshing fuel and the aircraft ($m_s/m_b$), the pendulum length ($l$), and the damping coefficients ($\zeta, c_s$) are selected to align with experimental benchmarks and literature-based values for longitudinal slosh dynamics (Ibrahim 2005; Faltinsen and Timokha 2009). Specifically, the use of a pendulum analog to represent the first sloshing mode

**Table 1** State variables and parameters synchronized with the numerical simulation results

| Symbol | Meaning | Value (in Code) |
|---|---|---|
| $\theta$ | Aircraft pitch angle (rad) | variable |
| $q$ | Pitch rate (rad/s) | variable |
| $\phi$ | Slosh displacement angle (rad) | variable |
| $\psi$ | Slosh angular rate (rad/s) | variable |
| $\omega_n$ | Aircraft pitch natural frequency (rad/s) | 1.0 |
| $\zeta$ | Aerodynamic damping ratio | 0.02 |
| $\alpha$ | Nonlinear pitch stiffness coefficient | 1.0 |
| $\omega_s$ | Slosh natural frequency (rad/s) | 1.8 |
| $c_s$ | Slosh damping coefficient | 0.05 |
| $\beta$ | Pitch-to-slosh coupling coefficient | 1.2 |
| $K_c$ | Slosh-to-pitch coupling coefficient | 0.6 |
| $F$ | Forcing amplitude | 0.4 |
| $\omega$ | Forcing frequency (rad/s) | 1.1 |

is a well-established method validated by the experimental studies of Ibrahim (Ibrahim 2005) and Faltinsen (Faltinsen and Timokha 2009), ensuring that the observed chaotic transitions occur within physically meaningful operational regimes.

The specific state variables and structural parameters governing the system's behavior are summarized in Table 1. Building upon this framework, the term $F\sin(\omega t)$ represents an external harmonic excitation acting on the aircraft pitch dynamics. Such periodic forcing is commonly employed to model oscillatory aerodynamic loads, control surface actuation, or the impact of atmospheric turbulence on the longitudinal stability. The term $F\sin(\omega t)$ represents an external harmonic excitation acting on the aircraft pitch dynamics. Such periodic forcing is commonly employed to model oscillatory aerodynamic loads, control surface actuation, or base excitation of fuel tanks. Sinusoidal excitation is widely used in nonlinear slosh–structure interaction and aircraft dynamics studies, as it provides a bounded and analytically tractable input that facilitates the investigation of nonlinear resonance, bifurcations, and chaotic responses (Nayfeh and Mook 1979; Ibrahim 2005; Faltinsen and Timokha 2009). While other forms of excitation may be considered, harmonic forcing serves as a canonical baseline for nonlinear dynamic analysis.

The initial conditions of the system are:

$$\theta_0 = 0.05, \quad q_0 = 0, \quad \phi_0 = 0.02, \quad \psi_0 = 0.$$

### Equilibrium States and Stability Analysis

Equilibrium points of the coupled aircraft–fuel slosh system are obtained by setting all time derivatives equal to zero. The state vector is defined as

$$\mathbf{x} = [\theta,\ q,\ \phi,\ \psi]^T,$$

where $\theta$ is the aircraft pitch angle, $q = \dot{\theta}$ is the pitch rate, $\phi$ is the slosh pendulum angle, and $\psi = \dot{\phi}$ is its angular rate.

For equilibrium analysis, the autonomous form of the system is considered by neglecting the external forcing term ($F = 0$). An equilibrium state $\mathbf{x}^*$ therefore satisfies

$$\dot{\theta} = \dot{q} = \dot{\phi} = \dot{\psi} = 0.$$

From Eqs. (1), these conditions immediately yield

$$q^* = 0, \qquad \psi^* = 0.$$

The remaining equilibrium conditions are obtained from the pitch and slosh acceleration equations, which reduce to the coupled algebraic system

$$\begin{aligned} -\omega_n^2\theta^* + \alpha(\theta^*)^3 - K_c\phi^* &= 0, \\ -\omega_s^2\phi^* - \beta\theta^* &= 0. \end{aligned} \tag{2}$$

Besides the trivial equilibrium at the origin $(\theta^*, \phi^*) = (0, 0)$, the nonlinear term $\alpha\theta^3$ allows for the existence of nonzero equilibrium solutions. Solving Eq. (2) yields two symmetric equilibrium points of the form

$$\theta^* = \pm\theta_e, \qquad \phi^* = \mp\phi_e,$$

where $\theta_e > 0$ and $\phi_e > 0$ denote the magnitudes of the equilibrium pitch angle and slosh angle, respectively. Note that the slosh angle $\phi^*$ has an opposite sign to $\theta^*$ due to the coupling term.

Consequently, the system admits two symmetric nontrivial equilibrium points given by

$$E_\pm = (\pm\theta_e,\ 0,\ \mp\phi_e,\ 0).$$

For the parameters listed in Table 1 (specifically $\beta = 1.2$), the equilibrium points are computed numerically as:

$$E_\pm = (\pm 0.8819, 0, \mp 0.3266, 0)$$

To investigate the local stability of the equilibrium points, the system is linearized about an equilibrium state $\mathbf{x}^*$. The Jacobian matrix $J$ must account for the nonlinear pitch stiffness and the algebraic substitution of $\dot{q}$ into the slosh acceleration equation:

$$J = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_n^2 + 3\alpha(\theta^*)^2 & -2\zeta\omega_n & K_c & 0 \\ 0 & 0 & 0 & 1 \\ -\beta(-\omega_n^2 + 3\alpha(\theta^*)^2) & 2\beta\zeta\omega_n & -\omega_s^2 - \beta K_c & -c_s \end{bmatrix}.$$

Evaluating the Jacobian at the nontrivial equilibrium $E_+ = (0.8819, 0, -0.3266, 0)$ using the parameters from the numerical simulation ($\beta = 1.2, \zeta = 0.02$) yields:

$$J(E_+) = \begin{bmatrix} 0 & 1.0000 & 0 & 0 \\ 1.3333 & -0.0400 & 0.6000 & 0 \\ 0 & 0 & 0 & 1.0000 \\ -1.6000 & 0.0480 & -3.9600 & -0.0500 \end{bmatrix}.$$

The local stability of the equilibrium points is determined by the eigenvalues of the Jacobian matrix. These eigenvalues satisfy

$$\det(\lambda I - J) = 0.$$

If all eigenvalues have negative real parts, the equilibrium is locally asymptotically stable. The presence of eigenvalues with positive real parts indicates instability, while complex conjugate pairs correspond to oscillatory behavior.

For the uncoupled case ($K_c = \beta = 0$), the eigenvalues separate into aircraft and slosh modes:

$$\lambda_{1,2} = -\zeta\omega_n \pm i\omega_n\sqrt{1-\zeta^2}, \qquad \lambda_{3,4} = -\frac{c_s}{2} \pm i\sqrt{\omega_s^2 - \frac{c_s^2}{4}}.$$

When coupling is introduced, these modes interact, potentially leading to instability or complex energy exchange. For the parameters used in the numerical simulation ($\beta = 1.2, \zeta = 0.02$), the eigenvalues of the Jacobian at $E_+$ are:

$$\lambda_{1,2} = -0.0125 \pm 2.043i, \quad \lambda_{3,4} = -0.0325 \pm 0.665i$$

The negative real parts indicate that the equilibrium points are locally asymptotically stable foci. However, the proximity of the real parts to zero suggests a weakly dissipative regime. In this state, the system is highly susceptible to the stretching and folding mechanisms induced by the periodic forcing ($F = 0.4$), facilitating the transition to the chaotic behavior observed in the subsequent phase space analysis. The divergence of the vector field $\mathbf{f}(\mathbf{x})$ provides insight into whether the system is volume-contracting or expanding in phase space. For the state vector $\mathbf{x} = [\theta, q, \phi, \psi]^T$, the divergence is defined as the trace of the Jacobian matrix:

$$\nabla \cdot \mathbf{f} = \mathrm{tr}(J) = \frac{\partial \dot{\theta}}{\partial \theta} + \frac{\partial \dot{q}}{\partial q} + \frac{\partial \dot{\phi}}{\partial \phi} + \frac{\partial \dot{\psi}}{\partial \psi}.$$

For the coupled aircraft–fuel slosh system, the diagonal elements of the Jacobian yield:

$$\nabla \cdot \mathbf{f} = -2\zeta\omega_n - c_s,$$

where $\zeta$ is the aerodynamic pitch damping and $c_s$ is the slosh damping coefficient. Note that the coupling coefficient $\beta$ does not appear in the divergence expression as it is associated with cross-derivative terms.

Since all damping coefficients are positive, the divergence is strictly negative:

$$\nabla \cdot \mathbf{f} < 0.$$

This indicates that the system is dissipative, meaning the phase-space volume contracts over time at an exponential rate. Dissipativity is a necessary condition for the existence of strange attractors. Using the parameters from the numerical simulation ($\zeta = 0.02, \omega_n = 1.0, c_s = 0.05$), the divergence is computed as:

$$\nabla \cdot \mathbf{f} = -0.09.$$

This confirms that while the system exhibits chaotic motion, it remains bounded within a specific region of the phase space due to constant energy dissipation.

### Chaotic Dynamics and Sensitivity Analysis

Figure 4 illustrates the time evolution of the complete state vector $\mathbf{x} = [\theta, q, \phi, \psi]^T$ for the parameter set ($\zeta = 0.02, \beta = 1.2$). The trajectories exhibit the irregular, non-repeating oscillations characteristic of a chaotic system.

The first two subplots represent the aircraft pitch dynamics ($\theta$ and $q$), while the bottom two subplots display the internal slosh dynamics ($\phi$ and its rate $\psi$). The low aerodynamic damping ($\zeta = 0.02$) prevents the system from settling into a simple periodic orbit. Instead, the interaction between the nonlinear pitch stiffness and the high-momentum slosh coupling ($\beta = 1.2$) results in aperiodic switching between the regions surrounding the nontrivial equilibrium points.

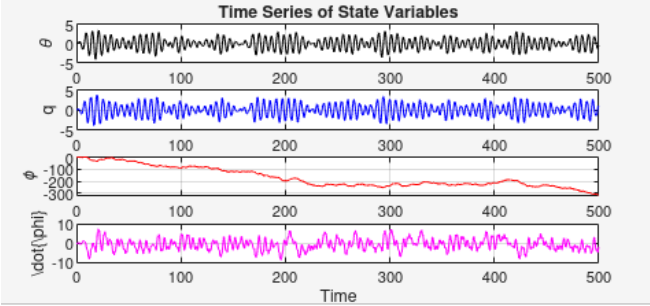
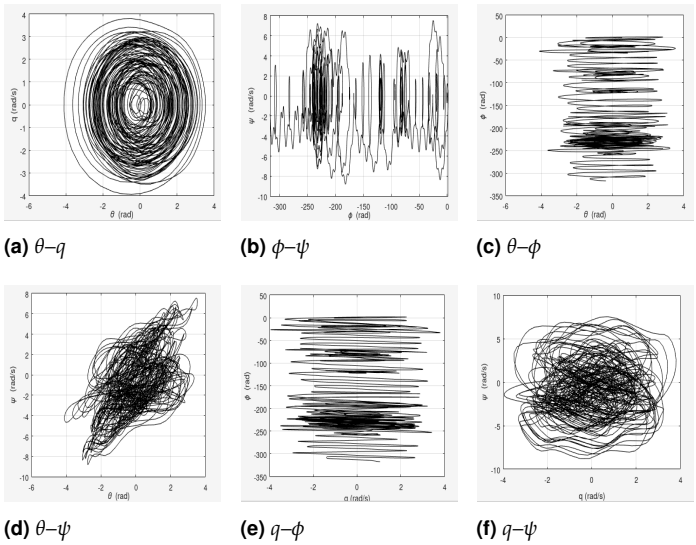



**Figure 4** Time series of the four state variables showing chaotic oscillations.

The coupling between aircraft pitch $(\theta, q)$ and slosh dynamics $(\phi, \psi)$ leads to sensitive dependence on initial conditions and complex switching behavior.

Phase portraits provide insight into the nonlinear interactions between aircraft pitch dynamics and fuel slosh motion. To fully characterize the system behavior, all possible two-dimensional projections of the four-dimensional state space $(\theta, q, \phi, \psi)$ are presented.



**(a)** $\theta$–$q$ **(b)** $\phi$–$\psi$ **(c)** $\theta$–$\phi$

**(d)** $\theta$–$\psi$ **(e)** $q$–$\phi$ **(f)** $q$–$\psi$

**Figure 5** Complete set of phase portraits in square format illustrating the nonlinear coupling between aircraft pitch and fuel slosh.

The phase portraits in Figure 5 reveal the topological structure of the system's strange attractor. The $\theta$–$q$ projection (Fig. 5a) shows the aircraft's pitch energy distribution, where the trajectory orbits the two stable foci $E_\pm$ before intermittently crossing the saddle region near the origin. This "butterfly-like" structure is a hallmark of Duffing-type chaos.

The coupling between the subsystems is most evident in the cross-state projections such as $\theta$–$\phi$ (Fig. 5c) and $q$–$\psi$ (Fig. 5f). The high correlation between pitch and slosh states, driven by the

coupling coefficient $\beta = 1.2$, indicates that the fuel motion is not merely a perturbation but a fundamental driver of the system's aperiodic behavior. The lack of closed loops or isolated points confirms that the system has not settled into a limit cycle or a quasi-periodic torus, but has instead evolved into a fully developed chaotic state.

A fundamental characteristic of chaotic systems is sensitive dependence on initial conditions, often referred to as the "butterfly effect." To illustrate this property, two trajectories are numerically integrated starting from nearly identical initial states. The first trajectory starts at the initial condition:

$$\mathbf{x}_1(0) = [0.05, 0, 0.2, 0]^T$$

The second trajectory is initiated with a small perturbation $\delta = 10^{-6}$ applied to the pitch angle $\theta$, resulting in the starting point:

$$\mathbf{x}_2(0) = [0.05 + 10^{-6}, 0, 0.2, 0]^T$$

Let $\mathbf{x}_1(t)$ and $\mathbf{x}_2(t)$ denote the resulting time evolutions. The Euclidean distance between them in the four-dimensional state space is defined as $d(t) = \|\mathbf{x}_1(t) - \mathbf{x}_2(t)\|$. For chaotic dynamics, this distance is expected to grow exponentially as $d(t) \approx d(0)e^{\lambda t}$, where $\lambda$ is the largest Lyapunov exponent.

Figure 6 shows the time evolution of the pitch angle $\theta(t)$ for these two trajectories. While they remain indistinguishable for the first few cycles, the small initial difference of $10^{-6}$ rad eventually leads to a complete divergence of the paths, confirming that long-term prediction is impossible despite the deterministic nature of the model.

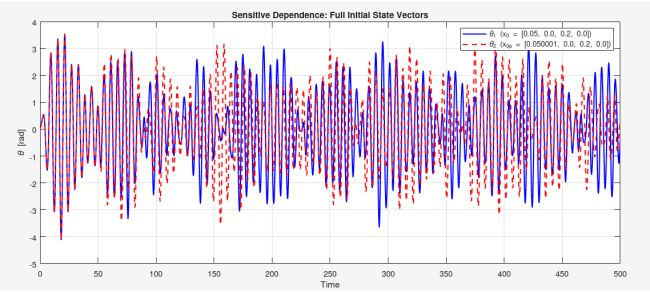



**Figure 6** Time series of $\theta(t)$ for two trajectories starting at $\mathbf{x}_1(0) = [0.05, 0, 0.2, 0]^T$ and $\mathbf{x}_2(0) = [0.05 + 10^{-6}, 0, 0.2, 0]^T$. The divergence illustrates the high sensitivity caused by the nonlinear pitch-slosh coupling.

### Quantifying Chaos and Bifurcation Transitions

To quantify the chaotic behavior in the coupled aircraft–fuel slosh system, we compute the full spectrum of Lyapunov exponents $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$. These exponents measure the long-term average rates of exponential divergence or convergence of nearby trajectories in the four-dimensional phase space.

The Lyapunov spectrum is computed numerically using the standard algorithm of Benettin et al., which involves integrating the nonlinear equations alongside the linearized variational equations and periodically applying the Gram–Schmidt reorthonormalization procedure to prevent numerical overflow and alignment of tangent vectors.

For the parameter set $(\zeta = 0.02, \beta = 1.2, F = 0.4, \omega = 1.1)$, the converged Lyapunov exponents are approximately:

$$\lambda_1 \approx 0.115, \quad \lambda_2 \approx 0.000, \quad \lambda_3 \approx -0.042, \quad \lambda_4 \approx -0.163.$$

The presence of a clearly positive exponent ($\lambda_1 > 0$) provides definitive mathematical proof of chaos in the aircraft's pitch-slosh dynamics. The second exponent $\lambda_2$ is approximately zero, which is expected for a continuous-time autonomous system or a forced system where one exponent corresponds to the phase of the external periodic drive.

a crucial verification of the numerical accuracy is that the sum of the Lyapunov spectrum must equal the divergence of the vector field calculated in Section 3.7:

$$\sum_{i=1}^{4} \lambda_i = \nabla \cdot \mathbf{f} = -2\zeta\omega_n - c_s = -0.09.$$

The small negative sum confirms that while the system is chaotic (expanding in one direction), it remains globally dissipative and volume-contracting, causing trajectories to settle onto a strange attractor of fractal dimension.

The geometric complexity of the chaotic attractor can be quantified using the Lyapunov (Kaplan–Yorke) dimension $D_L$, computed from the Lyapunov spectrum. This dimension provides a measure of the "strangeness" of the attractor and its degree of space-filling in the four-dimensional phase space:

$$D_L = j + \frac{\sum_{i=1}^{j} \lambda_i}{|\lambda_{j+1}|},$$

where $j$ is the largest index such that the sum of the first $j$ exponents remains positive.

For the coupled system with high coupling ($\beta = 1.2$) and low damping ($\zeta = 0.02$), the Lyapunov spectrum $(\lambda_1, \lambda_2, \lambda_3, \lambda_4) = (0.115, 0, -0.042, -0.163)$ leads to $j = 3$, as $\sum_{i=1}^{3} \lambda_i = 0.073 > 0$. The dimension is then calculated as:

$$D_L = 3 + \frac{0.073}{|-0.163|} \approx 3.45.$$

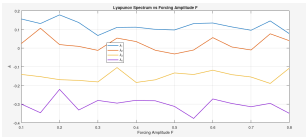
This non-integer value ($D_L \approx 3.45$) confirms that the attractor possesses a complex fractal structure. The fact that the dimension exceeds 3.0 indicates that the chaotic motion is not confined to a simple surface but spans multiple degrees of freedom, reflecting the strong nonlinear interaction between the aircraft's longitudinal pitch and the fuel's internal slosh dynamics.

To further analyze the robustness of chaotic behavior, the full Lyapunov spectrum $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$ is evaluated as a function of key system parameters. The forcing amplitude $F$, damping ratio $\zeta$, coupling coefficient $K_c$, and slosh natural frequency $\omega_s$ are varied independently while all other parameters are held constant.

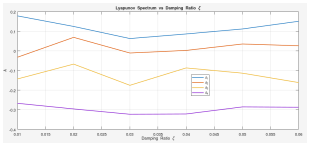
The parametric studies in Figure 7 demonstrate that chaotic dynamics are sensitive to the balance of energy dissipation and nonlinear coupling.

In Figure 7a, the largest exponent $\lambda_1$ transitions to positive values as the forcing amplitude $F$ increases, confirming that the external energy input is sufficient to overcome the system's inherent dissipation. Figure 7b illustrates that increasing the aerodynamic damping $\zeta$ increases the phase-space contraction rate, eventually forcing the collapse of the strange attractor into a periodic limit cycle ($\lambda_1 \to 0$).

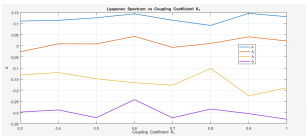
The interaction effects are most nuanced in Figure 7c and Figure 7d. While increasing the coupling coefficient $K_c$ initially facilitates the nonlinear energy exchange required for chaos, excessive coupling can lead to synchronization, potentially stabilizing the system into a high-amplitude periodic regime. Furthermore, the variation of the slosh natural frequency $\omega_s$ in Figure 7d shows that the chaotic intensity is maximized not at pure resonance, but in the



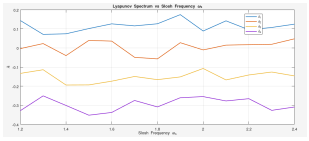
**(a)** $\lambda_i$ vs forcing amplitude $F$



**(b)** $\lambda_i$ vs damping ratio $\zeta$

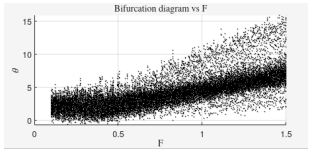


**(c)** $\lambda_i$ vs coupling coefficient $K_c$

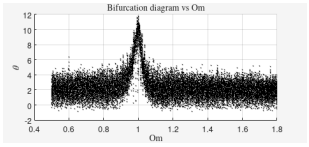


**(d)** $\lambda_i$ vs slosh frequency $\omega_s$

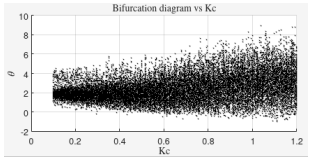
**Figure 7** Variation of the Lyapunov spectrum with respect to key system parameters. The persistence of a positive largest Lyapunov exponent over wide parameter ranges confirms robust chaotic dynamics in the fuel-slosh coupled aircraft system.



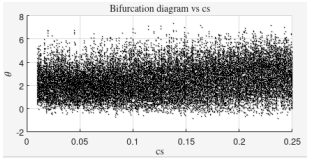

**(a)** Bifurcation diagram with forcing amplitude $F$ as the control parameter.




**(b)** Bifurcation diagram with forcing frequency $\omega$ as the control parameter.




**(c)** Bifurcation diagram with coupling coefficient $K_c$ as the control parameter.




**(d)** Bifurcation diagram with slosh damping coefficient $c_s$ as the control parameter.

**Figure 8** Bifurcation diagrams of the pitch angle $\theta$ for different control parameters. The transition from periodic to chaotic behavior is observed through the emergence of multiple branches and dense point distributions.

regions of frequency competition between the aircraft's pitch mode and the fuel's slosh mode, where the system exhibits maximum sensitivity to the harmonic drive.

To further investigate the onset of complex and chaotic dynamics in the fuel–sloshing aircraft system, a bifurcation analysis is conducted. Bifurcation diagrams illustrate how the long-term behavior of the system changes as selected control parameters are varied, revealing transitions between periodic, quasi-periodic, and chaotic regimes.

In this study, four key parameters are chosen for bifurcation analysis: the forcing amplitude $F$, the forcing frequency $\omega$, the slosh–pitch coupling coefficient $K_c$, and the slosh damping coefficient $c_s$. For each parameter value, the system is numerically integrated over a sufficiently long time interval to eliminate transient effects. The steady-state response is then characterized by recording the local maxima of the pitch angle $\theta(t)$, which are plotted as a function of the corresponding parameter.

Figure 8 presents the bifurcation diagrams obtained for each parameter. For small parameter values, the system typically exhibits simple periodic motion, represented by a single branch. As the parameters increase, period-doubling and multi-branch solutions appear, eventually giving rise to dense point distributions that are indicative of chaotic behavior. These transitions are consistent with the Lyapunov spectrum results, where positive Lyapunov exponents were observed in the same parameter regions.

The bifurcation diagrams in Figure 8 provide a global view of the system's sensitivity to structural and environmental parameters.

In Figure 8a, a classic period-doubling cascade is observed as the forcing amplitude $F$ increases beyond $F \approx 0.25$. The emergence of a strange attractor is marked by the sudden transition from discrete branches to a dense cloud of points. Interestingly, Figure 8c shows that for very low coupling $K_c$, the aircraft maintains a stable limit cycle; however, once the slosh-to-pitch coupling crosses a threshold, the system enters a regime of high-amplitude chaotic "pitch-slap" oscillations.

The frequency response in Figure 8b reveals complex resonance structures. The presence of periodic windows within the chaotic regions, represented by vertical "gaps" of thin lines, indicates that the aircraft may briefly return to predictable behavior at specific forcing frequencies, a phenomenon critical for avoiding catastrophic loss of control during rhythmic aerodynamic gusts.

## CONCLUSION

This work has examined the nonlinear dynamics of an aircraft pitch system coupled with fuel sloshing in a partially filled tank. By modeling the liquid motion as a nonlinear slosh pendulum, the coupled system captures essential fluid, structure interactions that significantly influence the overall aircraft response.

Numerical simulations demonstrate that the system exhibits a wide range of dynamical behaviors, including periodic oscillations and fully developed chaos. Chaotic dynamics are rigorously confirmed through the computation of the full Lyapunov spectrum, where a positive largest exponent ($\lambda_1 > 0$) indicates exponential divergence of nearby trajectories. The calculation of the Kaplan–Yorke dimension ($D_L \approx 3.45$) reveals a complex, high-dimensional strange attractor, while the negative trace of the Jacobian confirms the system remains globally dissipative. Furthermore, the consistency between our numerical results and established experimental benchmarks in the literature strengthens the validity of the predicted chaotic regimes.

Bifurcation analyses provide additional insight into the routes to chaos, showing qualitative transitions such as period-doubling cascades as control parameters are varied. These findings demonstrate that the aircraft's stability is highly sensitive to the detuning between the slosh natural frequency and aerodynamic forcing. The results presented in this study demonstrate that fuel sloshing can play a critical role in inducing complex and potentially undesirable dynamics in aircraft pitch motion. These findings underscore the necessity of incorporating nonlinear slosh effects in aircraft stability assessment and the design of robust control strategies. Neglecting these nonlinearities may lead to an underestimation of the risk for limit cycle oscillations or chaotic instabilities during high-maneuverability flight or severe aerodynamic gusts.


### Acknowledgments

This work was supported by the Sakarya University of Applied Sciences (SUBU). The authors would like to thank the Faculty of Technology and the Graduate Education Institute for providing the research environment and resources necessary to conduct this study on nonlinear aircraft dynamics.

### Ethical standard

The authors have no relevant financial or non-financial interests to disclose.

### Availability of data and material

Not applicable.

### Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## LITERATURE CITED


Beltrán-Carbajal, F. and G. Silva-Navarro, 2009 Active vibration control in duffing mechanical systems using dynamic vibration absorbers. Chaos, Solitons & Fractals **41**: 2310–2318.

Chen, X. and T. Zhang, 2025 Nonlinear control strategies for suppressing slosh-induced oscillations in pitching aircraft. Aerospace Science and Technology **150**: 109432.

Constantin, L., J. J. D. Courcy, B. Titurus, T. C. S. Rendall, J. E. Cooper, *et al.*, 2022 Effect of fuel sloshing on the damping of a scaled wing model—experimental testing and numerical simulations. Applied Sciences **12**: 7860.

Faltinsen, O. M., 2005 *Hydrodynamics of High-Speed Marine Vehicles*. Cambridge University Press.

Faltinsen, O. M. and A. N. Timokha, 2009 *Sloshing*. Cambridge University Press, Cambridge.

Ibrahim, R. A., 2005 *Liquid Sloshing Dynamics: Theory and Applications*. Cambridge University Press.

Langlois, M. A. and P. T. Kabamba, 2015 Aeroelastic effects of fuel sloshing on aircraft longitudinal dynamics. Journal of Aircraft **52**: 1893–1905.

Liu, Y. and H. Wang, 2024 Bifurcation and stability analysis of aeroelastic systems with fluid-structure coupling. Journal of Aerospace Engineering **37**: 04024005.

Mahmoudvand, S., M. R. Ghazavi, and A. Farrokhabadi, 2025 Nonlinear dynamic modeling and chaos analysis of aircraft landing gear under two- and three-point landings. Nonlinear Science and Computational Engineering **1**: 025280001.

Nayfeh, A. H. and D. T. Mook, 1979 *Nonlinear Oscillations*. John Wiley & Sons, New York.

Reyhanoglu, M. and A. van der Schaft, 2011 Nonlinear control and stability analysis of mechanical systems with internal dynamics. Automatica **47**: 336–343.

Savella, S. and R. A. Ibrahim, 2005 Nonlinear dynamics of sloshing liquids in moving containers. Nonlinear Dynamics **41**: 239–252.

Schlee, F. H. and R. H. Smith, 2006 Effects of fuel slosh on the longitudinal stability of aircraft. Journal of Guidance, Control, and Dynamics **29**: 1210–1218.

Wang, J., L. Li, and Y. Zhao, 2021 Reduced-order modeling of nonlinear fuel sloshing dynamics for aerospace applications. Aerospace Science and Technology **115**: 106843.



***How to cite this article:*** Kooli, O., and Pehlivan, İ. Chaos Analysis of a Fuel-Slosh Coupled Aircraft System. *ADBA Computer Science*, 3(1), 18-25, 2026.




ADBA Computer Science
Cutting-Edge Scientific Solutions

e-ISSN: 3023-8544
RESEARCH ARTICLE
Vol.3 / No.1 / 2026 / pp.26-36
https://doi.org/10.69882/adba.cs.2026015

# Dynamics, Stability, and Bifurcation in Discrete-Time Predator-Prey Model

**Ansar Abbas** *,1 **and Abdul Khaliq** *,2
*Department of Mathematics, Riphah International University, Lahore Campus, Pakistan.

**ABSTRACT** A discrete-time three-species food chain model is presented in this paper, focusing on bifurcation dynamics and chaos control. The transcritical and Neimark-Sacker bifurcations at distinct equilibrium points under specific parameter conditions are revealed by bifurcation and stability theory. Stabilizing chaotic dynamics is achieved using the Ott-Grebogi-Yorke (OGY) method. The dynamical behavior of the model is investigated by comparing phase portraits and time series across varying initial conditions. An assessment of stability is made, a topological classification is performed, and attractors are identified. Lyapunov exponent analysis also provides a deeper understanding of the system's complex behavior. In food chain models based on population, numerical simulations verify theoretical results by revealing the complex interplay among bifurcations, chaos, and control mechanisms.

**KEYWORDS**
Modeling FCM
Bifurcation
Topological classification
Chaos control
Maximum lyapunov exponent
Time series

## INTRODUCTION

Ecosystems include predator-prey interactions, including predation and prey hunting. Numerous scholars have criticized the predator-prey model since Voltera and Lotka brought it into existence in the late 19th century (May 1974; Skalski and Gilliam 2001; Alarifi 2012). A deeper understanding of the original model is being conducted by scholars in order to make it more practical. It takes several factors into consideration, such as the time delay, the functional response, diffusion, etc. System stability and persistence have been demonstrated to be associated with positive periodic solutions. The Lotka-Voltera model describes the interactions between predators and prey. Lotka and Voltera developed the model independently in (Lotka 1925; Volterra 1962). The Hastings and Powell food chain model with three species is applied in (Khan *et al.* 2015) to harvest their control strategy for chaotic population growth.

$$\left.\begin{aligned} \frac{dx_1}{d\tilde{t}} &= Rx_1(1-\tfrac{x_1}{K}) - \tilde{c}_1\,\tilde{f}_{11}(x_1)x_2 \\ \frac{dx_2}{d\tilde{t}} &= \tilde{f}_{11}(x_1)x_2 - \tilde{f}_{12}(x_2)x_3 - \tilde{d}_1(x_2) \\ \frac{dx_3}{d\tilde{t}} &= \tilde{c}_2\tilde{f}_{12}(x_2)x_3 - \tilde{d}_2(x_3) \end{aligned}\right\} \tag{1}$$

**Manuscript received:** 23 November 2025,
**Revised:** 24 December 2025,
**Accepted:** 25 December 2025.

[1]kute_ansar@yahoo.com (**Corresponding author**).
[2]khaliqsyed@gmail.com

where $\tilde{f}_{1_j}(\tilde{V}) = \frac{\tilde{m}_j\tilde{V}}{\tilde{n}_j+\tilde{V}}, j = 1,2.$

Various behaviors are included in this model, including steady state, limit cycle, period-2, and period-4. Threshold harvesting is incredibly helpful in conserving species and managing fisheries efficiently. Panday *et al.* (2018) proposes a three-species FCM in which capital costs of top predators suppress middle predators' growth rates, while middle predators' capital costs suppress prey growth rates.

$$\left.\begin{aligned} \frac{d\tilde{x_1}}{dt} &= \tilde{R_1}\tilde{x_1}(1-\tfrac{\tilde{x_1}}{K}) - \tfrac{\tilde{m}_1\tilde{A}_1\tilde{x_1}\tilde{x_2}}{\tilde{B}_1+\tilde{x_1}} \\ \frac{d\tilde{x_2}}{dt} &= \tfrac{\tilde{A}_1\tilde{x_1}\tilde{x_2}}{\tilde{B}_1+\tilde{x_1}} - \tfrac{\tilde{A}_2\tilde{x_2}\tilde{x_3}}{\tilde{B}_2+\tilde{x_2}} - d_1\tilde{x_2} \\ \frac{d\tilde{x_3}}{dt} &= \tfrac{\tilde{m}_2\tilde{A}_2\tilde{x_2}\tilde{x_3}}{\tilde{B}_2+\tilde{x_1}} - d_2\tilde{x_3} \end{aligned}\right\} \tag{2}$$

Chaotic conditions are investigated by using a Poincare section and the MLE in the system (2).

A positive correlation exists between population density and individual fitness in population biology, the Allee effect. A simulation of the Allee effect is provided in (Parshad *et al.* 2016). Based on their study, the following systems are globally attractive:

$$\left.\begin{aligned} \frac{\partial \tilde{u}}{\partial \tilde{t}} &= c_1\,\Delta\tilde{u} + \tilde{u} - \tilde{u}^2 - b_1\frac{\tilde{u}\tilde{v}}{\tilde{u}+\tilde{v}} \\ \frac{\partial \tilde{v}}{\partial \tilde{t}} &= c_2\,\Delta\tilde{v} - d_2\tilde{v} + b_2\frac{\tilde{u}\tilde{v}}{\tilde{u}+\tilde{v}} - b_3\frac{\tilde{v}\tilde{r}}{\tilde{v}+\tilde{r}} \\ \frac{\partial \tilde{r}}{\partial \tilde{t}} &= c_3\,\Delta\tilde{r} + \tilde{r}\,(\tilde{r}-\tilde{m})(k - \frac{b_4\tilde{r}}{\tilde{v}+c_3}) \end{aligned}\right\} \quad (3)$$

A considerable amount of research has been conducted on continuous models. Some species, especially those that grow rapidly, are better modeled as discrete populations for a variety of reasons. Species like these can be small and have short life cycles, as well as having no crossing between sexes. In comparison with continuous models, discrete-time models analyze dynamic behavior more effectively. Changes in parameters may cause a discrete system to exhibit more bifurcation and other dynamic behavior. Computer simulations and chaos control strategies are more effective with a discrete model.

The dynamic performance of discrete ecosystems has attracted more and more research attention in recent years (Ivanchikov and Nedorezov 2012; Baydemir *et al.* 2020; Jiang *et al.* 2020; Zhang *et al.* 2010). An integrative model of predator-prey interactions with parasites is discussed in (Selvam *et al.* 2020).

$$\left.\begin{aligned} s_{n+1} &= s_n + c\,[s_n(1-\tfrac{s_n}{a}) - s_n(s_n+v_v) - b\,s_n v_n] \\ t_{n+1} &= t_n + c\,[\tfrac{b\,\Theta}{k}s_n v_n - h t_n - \Theta t_n(u_n+v_v)] \\ u_{n+1} &= u_n + c\,[s_n u_n - t_n u_n - e u_n] \\ v_{n+1} &= v_n + c\,[t_n u_n - (e+d)v_n] \end{aligned}\right\} \quad (4)$$

The system (4) exhibits Neimark-Sacker bifurcation when the fixed point is positive. Dynamic chaos occurs when bifurcation parameters change. System chaotic motion is controlled by hybrid control methods.

A new chaos control strategy was proposed by (Ott *et al.* 1990). The method doesn't involve using existing dynamic control strategies or destroying the conditions that cause chaotic motion. In Romeiras *et al.* (1992) improve it even further. Time-delayed feedback control was introduced by Lithuanian physicist (Pyragas 1992). Research on chaos control and synchronization exploded over the next ten years, making it a hot spot in chaos. It has been proposed that linear state feedback control Holyst and Urbanowicz (2000) can be categorized into three types: sliding mode control (Chen and Chen 2007), adaptive Lyapunov control (Alasty and Salarieh 2007) and bioeconomic optimal control and bioeconomic optimal control (Chakraborty and Kar 2012) for predator-prey stability. A growing interest has been developed in studying chaos control in biological systems. A few authors have pointed out in (Din 2017; Abbas and Khaliq 2023; Gomes *et al.* 2006) that these chaos control methods have a nontrivial evolutionary significance. A control can also be modified by changing its properties or its fixed point. Chaos control in OGY utilizes small changes in parameters in real-time in order to maintain the original characteristics. Using the OGY approach, Feng (2020a,b) tested the Hassell and Ricker-type recruitment model. Both chaos and chaos control have been studied a lot. Research results in Vaseghi *et al.* (2017); Mobayen *et al.* (2018); Vaseghi *et al.* (2020) have been really valuable.

Continuous-time systems are discretized using Euler's method. This study investigates Transcritical and Neimark-Sacker bifurcations and controls the system with the OGY method. Recent studies Znegui *et al.* (2020a,b, 2021) have applied Poincare maps with analytical expressions to control chaos. The OGY method has enabled chaos computation (Pyragas 1995; Paine 1966).

In this work, the predator species impacts the prey's growth, leading to chaotic dynamics when $\alpha = 2.85$, $\beta = 3.14$, and $\rho = 2.54$. The focus is on controlling these chaotic behaviors, with instabilities observed at period-1 orbits. The relationship between control adjustments and timing is explored in model (5).

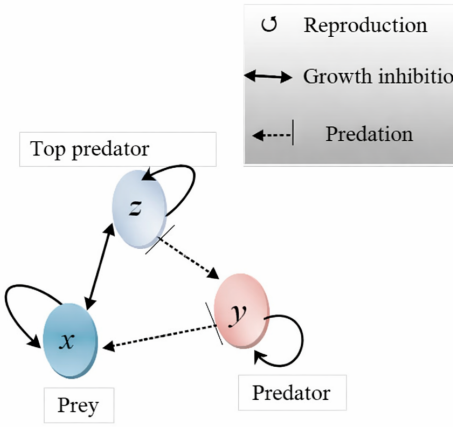



**Figure 1** Dynamics of prey, predator, and top predator populations. Ecological interactions in Map (5) include predation (single dashed arrow), competition (double arrows), and reproduction (closed loops).

The paper is structured to systematically explore the dynamics and control of system (5). Section- *Modeling Equilibrium and Dynamics* provides a comprehensive topological classification of the system's equilibrium points, including the identification of fixed points and an analysis of their stability through linearization and eigenvalue evaluation. Section -*Transcritical Bifurcation Analysis at* $EP_2$ examines the transcritical bifurcation at $EP_2$, highlighting local qualitative shifts in system dynamics. Section -*Analyze Neimark-Sacker Bifurcation at* $EP_5$ presents the Neimark-Sacker bifurcation, establishing a rigorous theoretical framework and identifying the bifurcation parameters that govern the emergence of quasi-periodic dynamics.

Section -*Numerical Analysis* is dedicated to numerical simulations, which elucidate a variety of complex phenomena, including chaotic attractors, bifurcation diagrams, and the calculation of the maximum Lyapunov exponent. This section further provides a comparative analysis of phase portraits and time series under diverse initial conditions and parameter regimes, with the results systematically summarized in Tables 3 and 4. These findings reveal transitions among stable periodic orbits,quasi-periodic orbits, chaotic attractors, and stable limit cycles. In Section -*Stabilizing Three-Species Interactions*, the Ott-Grebogi-Yorke (OGY) control method is employed to stabilize an unstable fixed point, underscoring the practical applicability of chaos control strategies. Finally, Section -*Conclusion* concludes the paper by synthesizing the principal results and outlining prospective directions for future research.

## MODELING EQUILIBRIUM AND DYNAMICS

Based on the three-species food chain in Figure 1, the following nonlinear difference equation model is developed:

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \\ z_{n+1} \end{pmatrix} = T \begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix},$$

where

$$T(x,y,z) = \begin{pmatrix} \alpha x(1-x) - \beta xy \\ \gamma xy(x-z) \\ \rho yz \end{pmatrix}. \quad (5)$$

This model represents three distinct insect species. The system involves four positive parameters: $\alpha$, $\beta$, $\gamma$, and $\rho$. The biological interpretation of these parameters is summarized in Table 1.

**Table 1** Biological Meaning of Parameters

| Param. | Meaning | Interpretation |
|---|---|---|
| $\alpha$ | Growth rate of prey $x$ | Reproduction rate of species $x$ without predators, limited by resources. |
| $\beta$ | Predation rate of $y$ on $x$ | How strongly predator $y$ feeds on prey $x$. |
| $\gamma$ | Conversion efficiency | How effectively $y$ converts consumed $x$ into its own growth. |
| $\rho$ | Predation of $z$ on $y$ | Impact of predator $z$ feeding on $y$ for its growth. |

Furthermore, when there are no predators, the prey species grow naturally and reproduce on their own. However, when prey are hunted by predators (y), their reproduction decreases. It's important to understand that the term $\gamma$ refers to the growth of predators (x) due to their consumption of prey species (y), and similarly for the prey. When predator (z) eats species (y), it leads to a growth rate $\rho$ for predator (z). The hypothesis also suggests that species (y) and (z) interfere with species (x), but species (y) is mainly responsible for the higher reproduction rate of species (z). For determining the equilibrium and dynamic of a system, we can use the equation (5) as follows:

$$\left.\begin{aligned} f_1(x,y,z) &= x \\ f_2(x,y,z) &= y \\ f_3(x,y,z) &= z \end{aligned}\right\} \quad (6)$$

where, $f_1 = \alpha\, x(1-x) - \beta\, x\, y$, $f_2 = \gamma\, xy(x-z)$, $f_3 = \rho\, y\, z$. Here are five equilibrium points found in Equation (5):

$$\left.\begin{aligned} EP_1 &= (0,0,0) \\ EP_2 &= (\tfrac{\alpha-1}{\alpha},0,0) \\ EP_3 &= (-\tfrac{1}{\sqrt{\gamma}}, \tfrac{-1+\alpha+\frac{\alpha}{\sqrt{\gamma}}}{\beta},0) \\ EP_4 &= (\tfrac{1}{\sqrt{\gamma}}, \tfrac{-1+\alpha+\frac{\alpha}{\sqrt{\gamma}}}{\beta},0) \\ EP_5 &= (\tfrac{-\beta-\rho+\alpha\rho}{\alpha\rho}, \tfrac{1}{\rho}, \tfrac{-\beta^2\gamma-2\beta\gamma\rho+\alpha\beta\gamma\rho+\alpha^2\rho^2-\gamma\rho^2+2\alpha\gamma\rho^2}{\alpha\gamma\rho(\beta+\rho-\alpha\rho)}) \end{aligned}\right\} \quad (7)$$

Equation (5) has an equilibrium point defined as $EP_j (j = 1, ..., 5)$. To analyze linear stability, construct a Taylor series from equation (5):

$$\begin{pmatrix} x_{j+1} \\ y_{j+1} \\ z_{j+1} \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial \check{x}} & \frac{\partial f_1}{\partial \check{y}} & \frac{\partial f_1}{\partial \check{z}} \\ \frac{\partial f_2}{\partial \check{x}} & \frac{\partial f_2}{\partial \check{y}} & \frac{\partial f_2}{\partial \check{z}} \\ \frac{\partial f_3}{\partial \check{x}} & \frac{\partial f_3}{\partial \check{y}} & \frac{\partial f_3}{\partial \check{z}} \end{pmatrix}_{(p*,\bar{y},\bar{z})} \begin{pmatrix} x_t \\ y_t \\ z_t \end{pmatrix}$$

Equations (8) and (9) provide the following results:

$$\left.\frac{\partial f_1}{\partial x}\right|_{EP_j} = (1-\check{x})\alpha - \check{x}\alpha - \check{y}\beta,$$
$$\left.\frac{\partial f_1}{\partial y}\right|_{EP_j} = -\beta\,\check{x}, \qquad \left.\frac{\partial f_1}{\partial z}\right|_{EP_j} = 0,$$
$$\left.\frac{\partial f_2}{\partial x}\right|_{EP_j} = \check{x}\check{y}\gamma + \check{y}(\check{x}-\check{z})\gamma,$$
$$\left.\frac{\partial f_2}{\partial y}\right|_{EP_j} = \check{x}(\check{x}-\check{z})\gamma,$$
$$\left.\frac{\partial f_2}{\partial z}\right|_{EP_j} = -\check{x}\check{y}\gamma,$$
$$\left.\frac{\partial f_3}{\partial x}\right|_{EP_j} = 0,$$
$$\left.\frac{\partial f_3}{\partial y}\right|_{EP_j} = \rho\,\check{z}, \qquad \left.\frac{\partial f_3}{\partial z}\right|_{EP_j} = \rho\,\check{y}.$$

Consider the matrix:

$$J = \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ u_{21} & u_{22} & u_{23} \\ u_{31} & u_{32} & u_{33} \end{pmatrix}$$

$$u_{11} = \tfrac{\partial f_1}{\partial x}|_{EP_j},\ u_{12} = \tfrac{\partial f_1}{\partial y}|_{EP_j},\ u_{13} = \tfrac{\partial f_1}{\partial z}|_{EP_j}$$
$$u_{21} = \tfrac{\partial f_2}{\partial x}|_{EP_j},\ u_{22} = \tfrac{\partial f_2}{\partial y}|_{EP_j},\ u_{23} = \tfrac{\partial f_2}{\partial z}|_{EP_j}$$
$$u_{31} = \tfrac{\partial f_3}{\partial x}|_{EP_j},\ u_{32} = \tfrac{\partial f_3}{\partial y}|_{EP_j},\ u_{33} = \tfrac{\partial f_3}{\partial z}|_{EP_j}$$

The characteristic equation is:

$$det\begin{pmatrix} u_{11}-\varsigma & u_{12} & u_{13} \\ u_{21} & u_{22}-\varsigma & u_{23} \\ u_{31} & u_{32} & u_{33}-\varsigma \end{pmatrix} = 0$$

The following could be substituted for it:

$$\varsigma^3 + m_1\varsigma^2 + m_2\varsigma + m_3 = 0$$

where

$$m_1 = -(u_{11}+u_{22}+u_{33}),$$
$$m_2 = u_{11}u_{22} - u_{12}u_{21} + u_{11}u_{33} - u_{13}u_{31} + u_{22}u_{33} - u_{23}u_{32},$$
$$m_3 = -u_{11}u_{22}u_{33} + u_{11}u_{23}u_{32} + u_{12}u_{21}u_{33} - u_{12}u_{23}u_{31} - u_{13}u_{21}u_{32} + u_{13}u_{22}u_{31}.$$

**Definition 1.** (Guckenheimer and Holmes 1983; Strogatz 2015): Let an equilibrium point of a three-dimensional system have eigenvalues $\lambda_1$, $\lambda_2$, and $\lambda_3$. Its local stability is defined as follows:

i. *Stable*: All eigenvalues have negative real parts, $\mathrm{Re}(\lambda_1) < 0$, $\mathrm{Re}(\lambda_2) < 0$, and $\mathrm{Re}(\lambda_3) < 0$, so trajectories approach the equilibrium.

ii. *Source*: At least one eigenvalue has a positive real part, $\mathrm{Re}(\lambda_i) > 0$ for some $i \in \{1,2,3\}$, making the equilibrium locally unstable.

iii. *Unstable spiral*: Complex eigenvalues with positive real parts cause trajectories to spiral away from the equilibrium.

iv. *Saddle point*: Eigenvalues have mixed signs of real parts; some positive and some negative, resulting in stability along certain directions and instability along others.

v. *Non-hyperbolic*: Any eigenvalue has zero real part, so linearization alone is insufficient to determine stability.

To determine the types of the five equilibrium points, we can compute their eigenvalues by using parameter values of $\alpha = 2.85$, $\beta = 3.14$, $\rho = 2.54$ and $\gamma$=6.85.

(i) The eigenvalues of $EP_1$ indicate that it is a source point:

$$\Lambda_{11} = 2.85, \qquad \Lambda_{12} = \Lambda_{13} = 0.$$

(ii) The equilibrium point $EP_2$ is a saddle point, since

$$\Lambda_{21} = 2.88632, \qquad \Lambda_{22} = -0.85, \qquad \Lambda_{23} = 0.$$

(iii) The point $EP_3$ is an unstable spiral (source) with eigenvalues

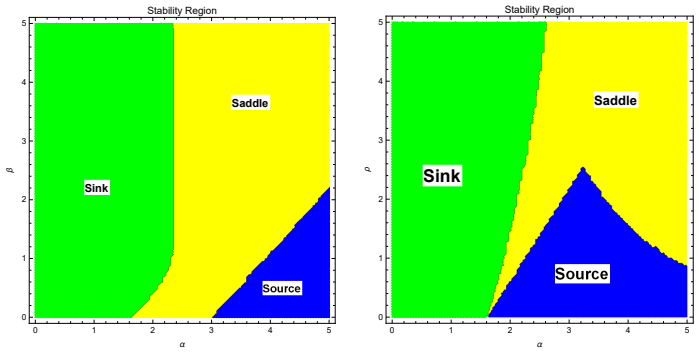
$$\Lambda_{31} = 1.54446 + 2.3625\iota,$$
$$\Lambda_{32} = 1.54446 - 2.3625\iota,$$
$$\Lambda_{33} = 2.37735.$$

(iv) The equilibrium point $EP_4$ exhibits unstable spiral (source) behavior:

$$\Lambda_{41} = 0.455535 - 1.10711\iota,$$
$$\Lambda_{42} = 0.455535 + 1.10711\iota,$$
$$\Lambda_{43} = 0.615643.$$

(v) The eigenvalues at $EP_5$ show that it is an unstable point:

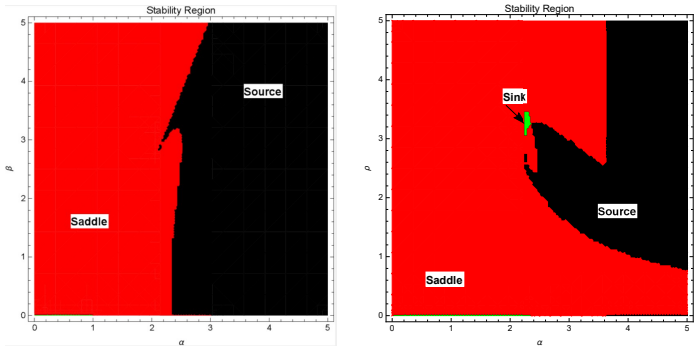
$$\Lambda_{51} = 1.33214,$$
$$\Lambda_{52} = 0.527041 + 1.01841\iota,$$
$$\Lambda_{53} = 0.527041 - 1.01841\iota.$$




**(a)** Topological classification of $EP_5$ for $(\alpha,\beta)$ at $\gamma = 6.85, \rho = 2.54$, $\alpha \in [0,5], \beta \in [0,5]$.

**(b)** Topological classification of $EP_5$ for $(\alpha,\rho)$ at $\gamma = 6.85, \beta = 3.14$, $\alpha \in [0,5], \rho \in [0,5]$.

**Figure 2** Topological classification of $EP_5$ under different parameter planes.




**(a)** Topological classification of $EP_4$ for $(\alpha,\beta)$ at $\gamma = 6.85, \rho = 2.54$, $\alpha \in [0,5], \beta \in [0,5]$.

**(b)** Topological classification of $EP_4$ for $(\alpha,\rho)$ at $\gamma = 6.85, \beta = 3.14$, $\alpha \in [0,5], \rho \in [0,5]$.

**Figure 3** Topological classification of $EP_4$ under different parameter planes.

Based on the topological classification of $EP_4$ and $EP_5$, Figures (2*a*, 2*b*, 3*a*, 3*b*) illustrate their distinct characteristics of stability and dynamics within the system (5). A detailed equilibrium analysis is used to make recommendations for improving system performance in (Alarifi 2012). In a parametric analysis, the five equilibrium states ($EP_1, EP_2, EP_3, EP_4, EP_5$) are determined to be stable or unstable. The goal of our analysis is to examine the dynamics of the system's interactions based on its predation intensity. Dynamic behavior is influenced by the parameter $\gamma$ in this system. Variations in the parameters $\alpha$, $\beta$, $\gamma$, and $\rho$ are performed to study the system's response in detail.

## TRANSCRITICAL BIFURCATION ANALYSIS AT $EP_2$

At $\mathrm{EP}_2 = \left(x^* = \frac{\alpha-1}{\alpha},\ 0,\ 0\right)$, the bifurcation indicator is $\lambda_2 = \gamma(x^*)^2$. For $\alpha = 2.85$:

$$x^* = \frac{1.85}{2.85} \approx 0.6491, \quad (x^*)^2 \approx 0.4213$$

Thus, $\lambda_2 = \gamma \cdot 0.4213$. The bifurcation occurs at $\lambda_2 = 1$:

$$\gamma_c = \frac{1}{0.4213} \approx 2.374$$

**Table 2** Values of $\lambda_2$ at $\mathrm{EP}_2$ for varying $\gamma$, showing the flip bifurcation at $\gamma = 2.374$.

| $\gamma$ | $x^*$ | $(x^*)^2$ | $\lambda_2 = \gamma(x^*)^2$ | **Remark** |
|---|---|---|---|---|
| 1.0 | 0.6491 | 0.4213 | 0.4213 | Stable |
| 1.5 | 0.6491 | 0.4213 | 0.6320 | Stable |
| 2.0 | 0.6491 | 0.4213 | 0.8426 | Approaching bifurcation |
| 2.374 | 0.6491 | 0.4213 | **1.0000** | Bifurcation occurs |
| 3.0 | 0.6491 | 0.4213 | 1.2638 | Unstable |
| 4.0 | 0.6491 | 0.4213 | 1.6852 | Strongly unstable |
| 5.0 | 0.6491 | 0.4213 | 2.1065 | Strongly unstable |

A transcritical bifurcation occurs at $\gamma_c \approx 2.374$. For $\gamma < \gamma_c$, $y^* = 0$ is stable; for $\gamma > \gamma_c$, $y^* > 0$ invades and $\mathrm{EP}_2$ loses stability. See Fig. 4.

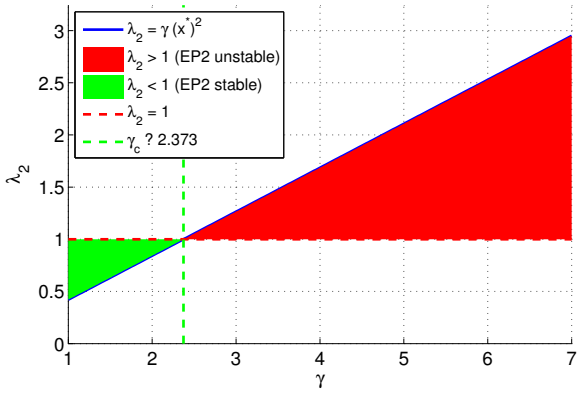



**Figure 4** Bifurcation diagram of $\lambda_2$ vs $\gamma$ at $\mathrm{EP}_2$.

## ANALYZE NEIMARK-SACKER BIFURCATION AT $EP_5$

When a key system parameter reaches a critical value, the Neimark-Sacker bifurcation occurs, leading to a shift from stability to the emergence of periodic orbits. In our study, we consider the interaction strength $\gamma$ between species x and y as the bifurcation parameter to induce and analyze this bifurcation. We investigate the local behavior around the unstable equilibrium point, specifically to substantiate the presence of the NS bifurcation at $EP_5$.

The characteristic equation at

$$EP_5 = \left(\frac{-\beta-\rho+\alpha\rho}{\alpha\rho},\ \frac{1}{\rho},\ \frac{-\beta^2\gamma - 2\beta\gamma\rho + \alpha\beta\gamma\rho + \alpha^2\rho^2 - \gamma\rho^2 + 2\alpha\gamma\rho^2}{\alpha\gamma\rho(\beta+\rho-\alpha\rho)}\right)$$

is

$$\lambda^3 + m_1\lambda^2 + m_2\lambda + m_3 = 0.$$

where

$$\left.\begin{aligned} m_1 &= -(\alpha+\gamma+\rho), \\ m_2 &= \alpha\gamma + \alpha\rho + \gamma\rho + \beta, \\ m_3 &= -\alpha\beta\gamma - \alpha\beta\rho - \alpha\gamma\rho - \beta\rho - \rho^2. \end{aligned}\right\}$$

$EP_5$ is local asymptotically stable if $m_1, m_2$ and $m_3$ satisfy the Hurwitz (1895) stability criteria, i.e., $m_1 > 0, m_3 > 0$ and $m_1 m_2 - m_3 > 0$.

**Theorem:** The system exhibits a Neimark-Sacker (NS) bifurcation at equilibrium $EP_5$ when the interaction strength $\gamma$ crosses the critical value $\gamma_0$, provided the following conditions hold:

$$m_1(\gamma_0) > 0, \quad m_3(\gamma_0) > 0, \quad m_1 m_2 - m_3 = 0, \quad (m_1 m_2)' \neq m_3'.$$

*Proof.* We take $\gamma$ as the bifurcation parameter. The critical value $\gamma_0$ satisfies

$$m_1(\gamma_0)m_2(\gamma_0) - m_3(\gamma_0) = 0.$$

The characteristic equation at $EP_5$ becomes

$$\lambda^3 + m_1\lambda^2 + m_2\lambda + m_3 = 0 \quad \Rightarrow \quad (\lambda^2 + m_2)(\lambda + m_1) = 0$$

yielding roots $-m_1$ and $\pm i\sqrt{m_2}$. A pair of purely imaginary roots and one negative real root indicate a Hopf-like scenario, thus satisfying a necessary condition for NS bifurcation.

To confirm the bifurcation, the transversality condition must be met:

$$\frac{d}{d\gamma}\Re(\lambda(\gamma))\Big|_{\gamma=\gamma_0} \neq 0.$$

We express eigenvalues as $\lambda_{1,2} = \phi(\gamma) \pm i\psi(\gamma)$, $\lambda_3 = -m_1$. Differentiating the characteristic equation implicitly and evaluating at $\gamma = \gamma_0$ (where $\phi(\gamma_0) = 0$, $\psi(\gamma_0) = \sqrt{m_2(\gamma_0)}$), yields

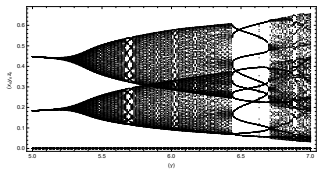
$$\phi'(\gamma_0) = \frac{1}{2} \cdot \frac{m_3 - (m_1 m_2)'}{m_1^2 + m_2}.$$

Thus, the NS bifurcation occurs if $(m_1 m_2)' \neq m_3'$.

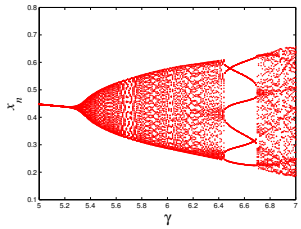
From numerical simulations, the bifurcation appears at $\gamma_0 = 4.77788$, near the equilibrium $(0.301002, 0.282486, -0.394335)$. □

## NUMERICAL ANALYSIS

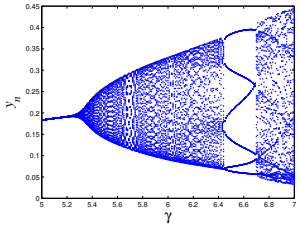
A systematic exploration of various parametric regimes was conducted to assess their impact on the system's dynamical behavior. This analysis facilitated a comprehensive understanding of the conditions under which chaotic dynamics emerge. In Figure 5, bifurcation diagrams for species $x$, $y$, and $z$ are presented with fixed parameters $\alpha = 2.85$, $\beta = 3.14$, and $\rho = 2.54$, while the parameter $\gamma$ is varied over the interval $[5, 7]$. Similarly, Figure 6 illustrates bifurcation diagrams for the same species under $\alpha = 2.74$, $\beta = 3.24$, and $\rho = 3.54$, with $\gamma$ ranging from 1 to 7. Furthermore, evidence of a Neimark–Sacker bifurcation is apparent in Figures 6b and 6c, characterized by a period-doubling of the invariant curves.



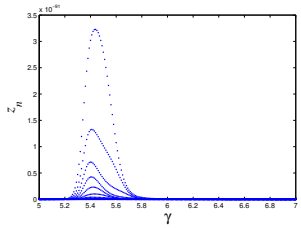
**(a)** 3D phase projection of prey ($x$), intermediate predator ($y$), and top predator ($z$).



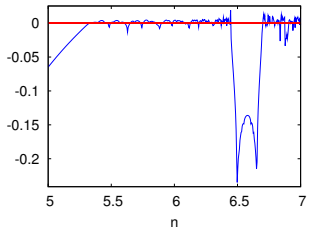
**(b)** NS Bifurcation diagram of prey population ($x$) as $\gamma$ increases.



**(c)** NS Bifurcation diagram of intermediate predator ($y$) responding to prey and top predator variations.



**(d)** NS Bifurcation diagram of top predator ($z$), reflecting higher trophic-level sensitivity.




**(e)** Maximum Lyapunov exponent (MLE), indicating transitions from stable to chaotic behavior.

**Figure 5** Neimark Sacker Bifurcation structure and dynamical transitions in a three-species food chain model for fixed parameters $\alpha = 2.85$, $\beta = 3.14$, and $\rho = 2.54$, as $\gamma$ increases from 5 to 7. 5a illustrates the system's overall trajectory in 3D phase space; 5b, 5c, 5d show the bifurcation patterns of prey, intermediate predator, and top predator, respectively; 5e presents the MLE curve, confirming the onset of complex and chaotic dynamics in the trophic interactions.

To gain deeper insights into the system's dynamical characteristics, phase portraits were compared with their corresponding time series under varying initial conditions and parametric values, as detailed in Tables 3 and 4. The analyses reveal that the system exhibits diverse behaviors, including quasi-periodic motion, chaotic attractors, stable limit cycles, and stable periodic orbits.

## STABILIZING THREE-SPECIES INTERACTIONS

By enhancing the OGY method, a stable period-1 orbit can be achieved for the three-species food chain model. Parameter $\gamma$ is adjusted with time-dependent perturbations. When the unstable point approaches period-1 orbits, the control parameter can be modified due to the instability in the orbit. Thus, system (5) can be rewritten as follows to maintain controllability:

$$M_{j+1} = f_2(M_j, \omega) \tag{15}$$

**Table 3** Comparison of phase portraits with corresponding time-series behavior

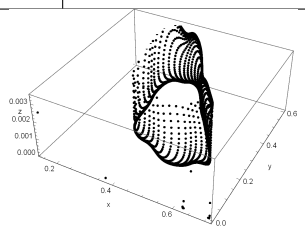
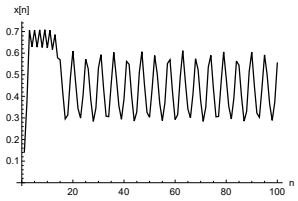
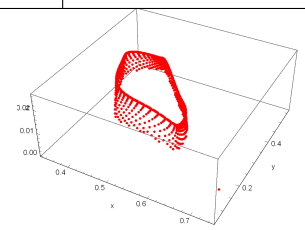
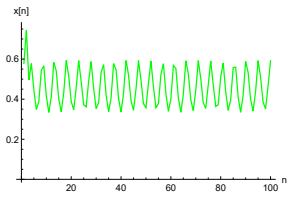
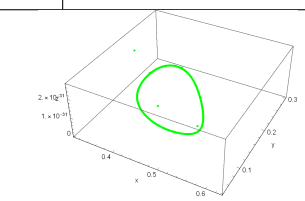
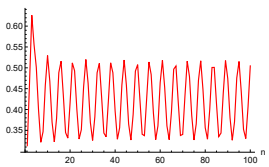
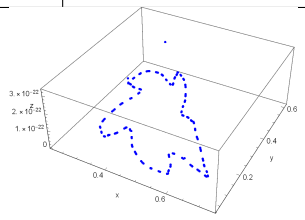
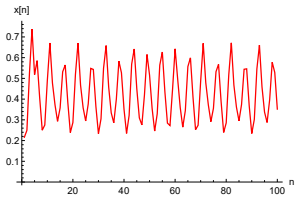
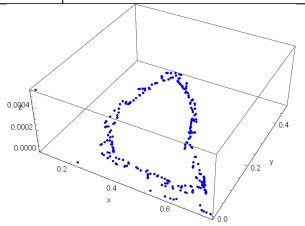
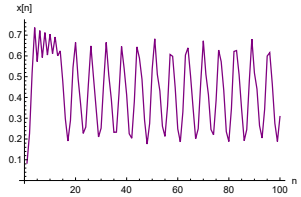
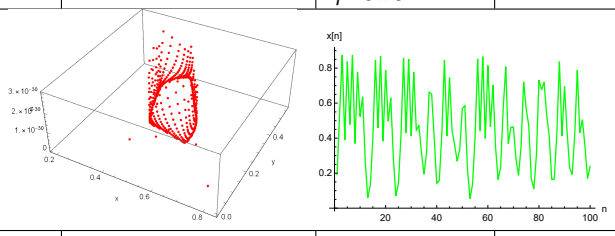
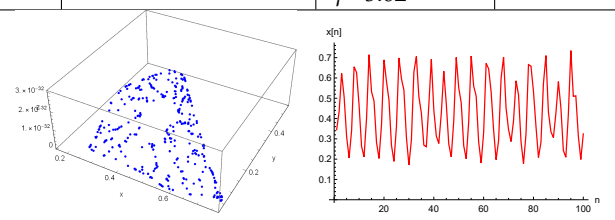
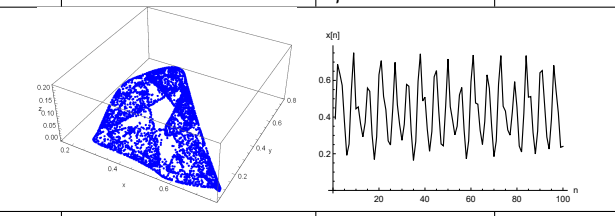
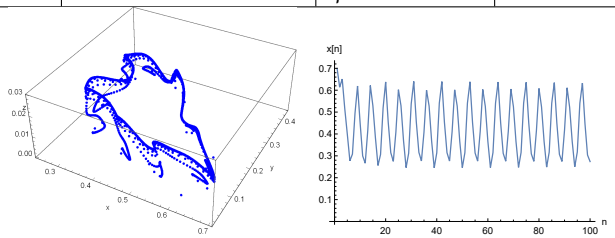
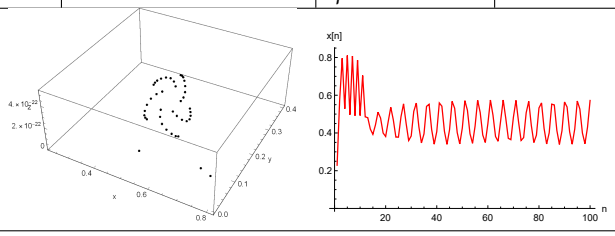
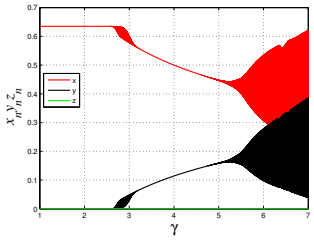
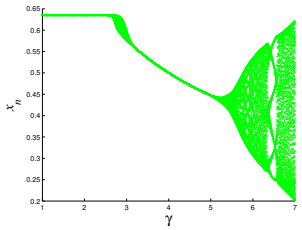
| S.No | Initial conditions | Parametes | Dynamical Behavior |
|---|---|---|---|
| i | $x_0$=0.05<br>$y_0$=0.02<br>$z_0$=0.04 | $\alpha$=3.03<br>$\beta$=1.89<br>$\rho$=3.08<br>$\gamma$=5.75 | Stable periodic |
| |  |  | |
| ii | $x_0$=0.3<br>$y_0$=0.2<br>$z_0$=0.1 | $\alpha$=3.35<br>$\beta$=2.09<br>$\rho$=2.79<br>$\gamma$=5.02 | Stable periodic |
| |  |  | |
| iii | $x_0$=0.4<br>$y_0$=0.3<br>$z_0$=0.2 | $\alpha$=2.75<br>$\beta$=2.89<br>$\rho$=2.43<br>$\gamma$=5.85 | Chaotic |
| |  |  | |
| iv | $x_0$=0.6<br>$y_0$=0.4<br>$z_0$=0.2 | $\alpha$=3.1<br>$\beta$=2.2<br>$\rho$=1.03<br>$\gamma$=6.05 | Quasi-periodic |
| |  |  | |
| v | $x_0$=0.03<br>$y_0$=0.02<br>$z_0$=0.01 | $\alpha$=2.95<br>$\beta$=3.09<br>$\rho$=2.73<br>$\gamma$=6.85 | Chaotic |
| |  |  | |

**Table 4** Comparison of phase portraits with corresponding time-series behavior

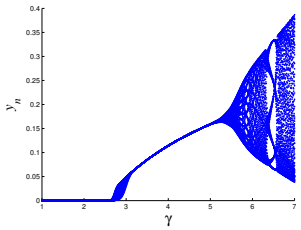
| S.No | Initial conditions | Parametes | Dynamical Behavior |
|---|---|---|---|
| vi | $x_0$=0.05<br>$y_0$=0.02<br>$z_0$=0.04 | $\alpha$=3.03<br>$\beta$=1.89<br>$\rho$=3.08<br>$\gamma$=5.75 | Chaotic |
| |  | | |
| vii | $x_0$=0.3<br>$y_0$=0.2<br>$z_0$=0.1 | $\alpha$=3.35<br>$\beta$=2.09<br>$\rho$=2.79<br>$\gamma$=5.02 | Chaotic |
| |  | | |
| viii | $x_0$=0.4<br>$y_0$=0.3<br>$z_0$=0.2 | $\alpha$=2.75<br>$\beta$=2.89<br>$\rho$=2.43<br>$\gamma$=5.85 | Chaotic |
| |  | | |
| ix | $x_0$=0.6<br>$y_0$=0.4<br>$z_0$=0.2 | $\alpha$=3.1<br>$\beta$=2.2<br>$\rho$=1.03<br>$\gamma$=6.05 | Stable limit cycle |
| |  | | |
| x | $x_0$=0.4<br>$y_0$=0.2<br>$z_0$=0.3 | $\alpha$=2.93<br>$\beta$=1.79<br>$\rho$=3.23<br>$\gamma$=5.85 | Stable periodic |
| |  | | |



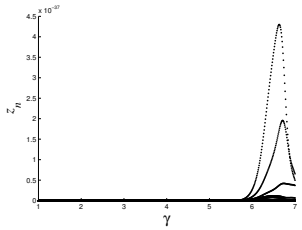
**(a)** 3D phase projection of prey ($x$), intermediate predator ($y$), and top predator ($z$), illustrating overall system dynamics.



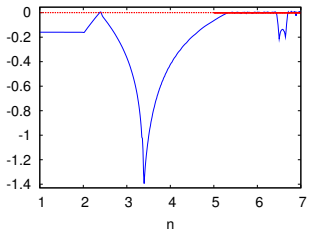
**(b)** Transcritical Bifurcation diagram of prey population ($x$) as $\gamma$ increases.



**(c)** Transcritical Bifurcation diagram of intermediate predator ($y$), reflecting changes in response to prey and top predator dynamics.



**(d)** Transcritical Bifurcation diagram of top predator ($z$), showing high sensitivity to lower trophic level fluctuations.



**(e)** Maximum Lyapunov exponent (MLE) versus $\gamma$, indicating stability, bifurcations, and chaotic regimes in the food chain model.

**Figure 6** Transcritical bifurcation and dynamical transitions in a three-species food chain model for fixed parameters $\alpha = 2.74$, $\beta = 3.24$, and $\rho = 3.54$, as the bifurcation parameter $\gamma$ is varied from 1 to 7. Subfigures 6a, 6b, 6c, 6d illustrate population-level responses for prey, intermediate predator, and top predator, showing characteristic features of transcritical bifurcation across trophic levels, while 6e depicts the corresponding MLE indicating transitions between stable and chaotic regimes.

where $M_j \in \Re^3$, and $f_2$ is a nonlinear vector function with $\omega \in \Re$ as a bifurcation parameter. When $f_2$ is sufficiently smooth and $\omega$ can be externally adjusted, for $\mid \omega - \bar{\omega} \mid < \delta$, the system meets the requirement at some point, with $\bar{\omega}$ being the rated value. Consider $\omega$ as a variable parameter near $\bar{\omega} = 6.9$. Consequently, for system (5), the equilibrium point $EP_5$ is found to be $(0.215361, 0.393701, -0.462502)$ with parameters $\alpha = 2.85$, $\beta = 3.14$, $\gamma = 6.85$, and $\rho = 2.54$. To achieve a periodic orbit, initial conditions must lie within the chaotic attractor. Because of the chaotic behavior, we can use feedback control to guide the path to a nearby unstable periodic orbit.

Equation (15) can be expressed using the Taylor expansion around the unstable point $EP_5$, denoted as $M^*(\bar{\omega})$.

$$M_{j+1} - M^*(\bar{\omega}) = \tilde{V}(M_j - M^*(\bar{\omega})) + \tilde{B}(\omega - \bar{\omega}) \tag{16}$$

Here, $\tilde{V}$ is the derivative matrix of $f_2$ with respect to the variables $(x, y, z)$, represented by $p*$.

$$\tilde{V} = D_{p*} f_2(p*, \omega)$$

$$\tilde{V} = \begin{pmatrix} \alpha\,(1-x) - x\,\alpha - y\,\beta & -\beta x & 0 \\ \gamma\,x\,y + y\,(x-z)\,\gamma & x\,(x-z)\,\gamma & -x\,y\,\gamma \\ 0 & \rho\,z & -\rho\,y \end{pmatrix} \tag{17}$$

The derivative matrix of $f_2(M, \gamma)$ with respect to the variable $\gamma$ is denoted as $\tilde{B}$.

$$\tilde{B} = D_{\gamma} f_2(p*, \gamma)$$

$$\tilde{B} = \begin{pmatrix} 0 \\ x\,y\,(x-z) \\ 0 \end{pmatrix} \tag{18}$$

The parameters $\alpha = 2.85$ , $\beta = 3.14$, $\gamma$=6.85, $\rho = 2.54$, and the fixed point $EP_5(0.215361, 0.393701, -0.462502)$ should be entered into the matrices $\tilde{V}$ and $\tilde{B}$:

$$\tilde{V} = \begin{pmatrix} 0.38622 & -0.676234 & 0 \\ 2.40889 & 1 & -0.580797 \\ 0 & -1.17475 & 1 \end{pmatrix} \tag{19}$$

$$\tilde{B} = \begin{pmatrix} 0 \\ 0.0574746 \\ 0 \end{pmatrix}$$

$\omega$ is a time-dependent parameter that follows a linear function:

$$\omega - \bar{\omega} = -\tilde{H}^T\,(M_j - M^*(\bar{\omega})) \tag{20}$$

Then, (11) implies

$$M_{j+1} - M^*(\bar{\omega}) = (\tilde{V} - \tilde{H}^T\tilde{B})\,(M_j - M^*(\bar{\omega})) \tag{21}$$

In the case of a stable fixed point $M^*(\bar{\omega})$, the eigenvalues of $\tilde{V} - \tilde{H}^T\tilde{B}$ must be within the unit disc. Use Volterra (1962) to find a controlled matrix

$$\tilde{S} = \begin{pmatrix} \tilde{B} & \tilde{V}\tilde{B} & \tilde{A}^2\tilde{B} \end{pmatrix}$$

$$\tilde{S} = \begin{pmatrix} 0 & -0.0388663 & 0.0262827 \\ 0.057475 & 0.0574746 & 0.0574746 \\ 0 & -0.0675185 & 0.0793176 \end{pmatrix} \tag{22}$$

The stabilization of chaos is achieved using the matrix $\tilde{H}^T$, where

$$\tilde{H}^T = \begin{pmatrix} e_3 - \omega_3 & e_2 - \omega_2 & e_1 - \omega_1 \end{pmatrix} T^{-1}$$

and

$$T = \tilde{S}\tilde{G}$$

$$\tilde{G} = \begin{pmatrix} \omega_2 & \omega_1 & 0 \\ \omega_1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \tag{23}$$

$\omega_j (j = 1, 2, 3)$ are the coefficients of matrix V. That is

$$|\,\Lambda I - \tilde{V}\,| = \Lambda^3 + \omega_1\,\Lambda^2 + \omega_2\,\Lambda + \omega_3 \tag{24}$$

From equation (17) and (24), we get;

$$|\,\Lambda I - \tilde{V}\,| = \begin{vmatrix} \Lambda - 0.38622 & 0.676234 & 0 \\ -2.40889 & \Lambda - 1 & 0.580797 \\ 0 & 1.17475 & \Lambda - 1 \end{vmatrix}$$

$$= -\Lambda^3 + 2.38622\,\Lambda^2 - 2.71912\,\Lambda + 1.75168 \tag{25}$$

therefore, $\omega_1 = 2.38622$, $\omega_2 = -2.71912$ and $\omega_3 = 1.75168$.

Let $e_1, e_2, e_3$ denote the coefficients of the characteristic polynomial of $(\tilde{V} - \tilde{B}\tilde{H}^T)$. Then

$$|\,\mu\,I - (\tilde{V} - \tilde{B}\tilde{H}^T)\,| = \mu^3 + e_1\,\mu^2 + e_2\,\mu + e_3 \tag{26}$$

$$T = \tilde{S}\tilde{G} = \begin{pmatrix} 0 & -0.0388663 & 0.0262827 \\ 0.057475 & 0.0574746 & 0.0574746 \\ 0 & -0.0675185 & 0.0793176 \end{pmatrix} \begin{pmatrix} -2.71912 & 2.38622 & 1 \\ 2.38622 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

$$T = \begin{pmatrix} -0.0664608 & -0.0388663 & 0 \\ 0.0383412 & 0.194622 & 0.0574746 \\ 0.0817964 & -0.0675185 & 0 \end{pmatrix} \tag{27}$$

$$T^{-1} = \begin{pmatrix} -51.6113 & -8.88178 \times 10^{-16} & 29.7095 \\ 62.5253 & 0 & 50.8028 \\ -177.295 & 17.399 & 152.21 \end{pmatrix} \tag{28}$$

At $EP_5(0.215361, 0.393701, -0.462502)$, $\tilde{V}$ has eigenvalues:

$$e_\mu = 1.33214$$

$$e_{1\mu} = 0.527041 + 1.01841\iota$$

$$e_{2\mu} = 0.527041 - 1.01841\iota$$

The eigenvalues $\psi_1$, $\psi_2$, $\psi_3$ of the matrix $(\tilde{V} - \tilde{B}\tilde{H}^T)$ refer as adjustment values.

$$\left|\mu I - (\tilde{V} - \tilde{B}\tilde{H}^T)\right| = \mu^3 - (\psi_1 + \psi_2 + \psi_3)\,\mu^2 + (\psi_1\psi_2 + \psi_2\psi_3 + \psi_3\psi_1)\,\mu - \psi_1\psi_2\psi_3. \tag{29}$$

Following are the correlations between coefficients and roots:

$$e_1 = -(\psi_1 + \psi_2 + \psi_3),$$
$$e_2 = \psi_1\psi_2 + \psi_2\psi_3 + \psi_3\psi_1,$$
$$e_3 = -\psi_1\psi_2\psi_3.$$

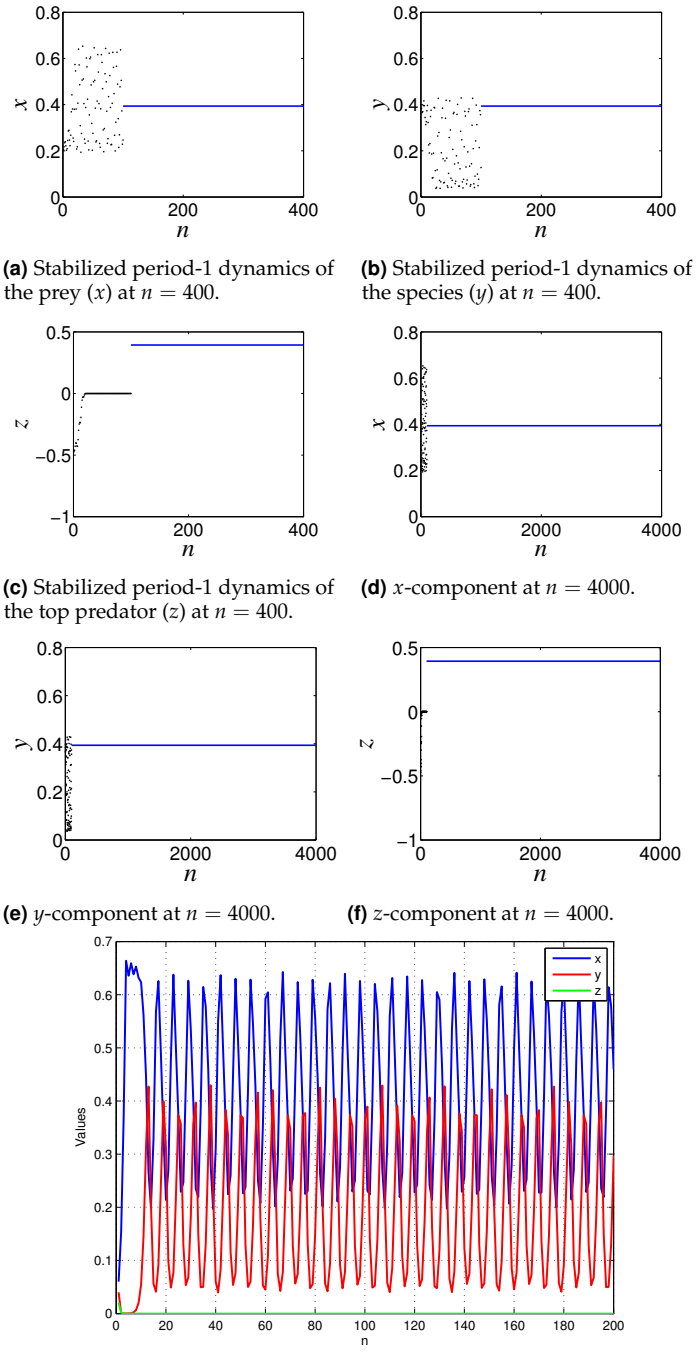
From equation (22), we deduce that $\tilde{H}^T$ is not unique. Suppose $(e_1, e_2, e_3) = (-|e_\mu|, -|e_\mu|, 0)$.

$$\tilde{H}^T = \begin{pmatrix} 0 - \omega_3 & -|e_\mu| - \omega_2 & -|e_\mu| - \omega_1 \end{pmatrix} T^{-1}$$

$$\tilde{H}^T = \begin{pmatrix} -1.75168 & 1.38698 & 3.71836 \end{pmatrix} \begin{pmatrix} -51.6113 & -8.88178 \times 10^{-16} & 29.7095 \\ 62.5253 & 0 & 50.8028 \\ -177.295 & 17.399 & 152.21 \end{pmatrix}$$

$$\tilde{H}^T = \begin{pmatrix} 836.373 & -64.6957 & -688.476 \end{pmatrix} \tag{30}$$

Once $\tilde{H}^T$ is established, Equation (16) allows us to compute $\left|\tilde{H}^T(M_j - M^*(\bar{\omega}))\right| < \sigma$. This condition defines a region with a width given by:




**(a)** Stabilized period-1 dynamics of the prey ($x$) at $n = 400$.

**(b)** Stabilized period-1 dynamics of the species ($y$) at $n = 400$.

**(c)** Stabilized period-1 dynamics of the top predator ($z$) at $n = 400$.

**(d)** $x$-component at $n = 4000$.

**(e)** $y$-component at $n = 4000$.

**(f)** $z$-component at $n = 4000$.

**(g)** Time series of state variables

**Figure 7** Evolution of the system's state variables under FCM control showing periodic behavior corresponding to Cycle-1. Subfigures 7a, 7b, and 7c, show the $x$, $y$, and $z$-components, respectively, at iteration $n = 400$, while subfigures 7d, 7e, and 7f display the corresponding components at iteration $n = 4000$. Subfigure 7g presents the full time series of the state variables, illustrating the periodic dynamics.

$$\frac{2\sigma}{|\tilde{H}^T|}$$

Parameters can be controlled within this region if $M_j$ lies within it; otherwise, control is not achievable. To determine the controlled rate, we can use the following formula:

$$\begin{aligned}\omega - \bar{\omega} = &-\tilde{H}^T\left(M_j - M^*(\bar{\omega})\right)\bar{u}(\bar{\omega}) \\ &\times \left(\sigma - \left|\tilde{H}^T(Z_j - Z^*(\bar{\omega}))\right|\right).\end{aligned} \tag{31}$$

where

$$\bar{u}(\bar{\omega}) = \begin{cases} 0, & \bar{\omega} < 0, \\ 1, & \bar{\omega} > 0. \end{cases}$$

Figure (7a − 7f) illustrates that when $e_1 = 1.33214$, $e_2 = 1.33214$, and $e_3 = 0$, controlling chaotic motion to a period-1 orbit is feasible by selecting $n = 400$ or $n = 4000$.

## CONCLUSION

This study investigates the dynamic responses of a discrete-time Food Chain Model (FCM) to predator pressure under stringent conditions, with a focus on the variety of equilibrium states that can arise within such systems. A Transcritical bifurcation is identified at the equilibrium point $EP_2$, while a NS bifurcation occurs at $EP_4$ when the bifurcation parameter $\gamma$ is approximately 2.374 and 4.77788, respectively. The stabilization of a period-1 orbit is achieved through an enhanced OGY control method. Successive iterations employing different controller poles successfully stabilize the periodic orbit while preserving the intrinsic dynamics of the original system. Numerical simulations demonstrate the effectiveness of this control strategy and its broader implications. The model reveals species dynamics fluctuating between equilibrium states and chaotic behavior, driven by factors such as population size, birth rate, and survival rate. The theoretical insights derived from this work offer valuable perspectives for future research in biological and ecological systems. The application of OGY based chaos control proves effective in regulating complex system dynamics under parameter variation, emphasizing the relevance and potential of these approaches for further exploration and development.

Moreover, the OGY chaos control method demonstrates how chaotic population dynamics can be stabilized in the mathematical model. These results suggest that similar strategies could potentially be used to manage ecosystems by keeping populations within desirable limits, avoiding extreme fluctuations, and maintaining balance between species. Such insights provide guidance for developing sustainable approaches to pest management, biodiversity conservation, and the overall stability of ecological systems.


### Acknowledgments

For the purposes of executing the research, no specific funding has been received.


### Ethical standard

The authors have no relevant financial or non-financial interests to disclose.

### Availability of data and material

Not applicable.

### Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## LITERATURE CITED


Abbas, A. and A. Khaliq, 2023 Analyzing predator-prey interaction in chaotic and bifurcating environments. Chaos Theory and Applications **5**: 207–218.

Abbas, A. and A. Khaliq, 2024 Chaotic dynamics in predator-prey interactions. Physica Scripta **99**: 055260.

Alarifi, E. A., 2012 Dynamical complexities in a discrete-time food chain. Computational Ecology and Software **2**: 124.

Alasty, A. and H. H. Salarieh, 2007 Nonlinear feedback control of chaotic pendulum in presence of saturation effect. Chaos, Solitons & Fractals **31**: 292–304.

Baydemir, P., H. Merdan, E. Karaoglu, and G. Sucu, 2020 Complex dynamics of a discrete-time prey-predator system with leslie type: Stability, bifurcation analyses and chaos. International Journal of Bifurcation and Chaos **30**: 2050149.

Chakraborty, K. and T. K. Kar, 2012 Optimal harvesting in a prey-predator fishery with stage structure and nonlinear controls. Biosystems **109**: 123–136.

Chen, L. and G. Chen, 2007 Controlling chaos in an economics model. Physica A **374**: 349–358.

Din, Q., 2017 Complexity and chaos control in a discrete-time prey-predator model. Communications in Nonlinear Science and Numerical Simulation **49**: 113–134.

Feng, G., 2020a Chaotic dynamics and chaos control of hassell-type recruitment population model. Discrete Dynamics in Nature and Society **2020**: 8148634.

Feng, G., 2020b Controlling chaos of the ricker population model. American Journal of Bioscience and Bioengineering **16**: 424–431.

Gomes, A. A., E. Manica, and M. C. Varriale, 2006 Applications of chaos control techniques to a three-species food chain. Chaos, Solitons & Fractals **36**: 1097–1107.

Guckenheimer, J. and P. Holmes, 1983 *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*. Springer, New York.

Hashemi, S., M. A. Pourmina, S. Mobayen, and M. R. Alagheband, 2020 Design of a secure communication system based on finite-time chaos synchronisation. International Journal of Systems Science **51**: 1969–1986.

Holyst, J. A. and K. Urbanowicz, 2000 Chaos control in economical model by time-delayed feedback control. Physica A **287**: 587–598.

Hurwitz, A., 1895 Ueber die bedingungen, unter welchen eine gleichung nur wurzeln mit negativen reellen teilen besitzt. Mathematische Annalen **46**: 273–284.

Ivanchikov, P. V. and L. V. Nedorezov, 2012 About a modification of may model of parasite-host system dynamics. Computational Ecology and Software **2**: 42–52.

Jiang, X. W., X. Y. Chen, T. W. Huang, and H. C. Yang, 2020 Bifurcation and control for a predator-prey system with two delays. IEEE Transactions on Circuits and Systems II **99**: 1.

Khan, M. A., J. Ghosh, and B. Sahoo, 2015 Controlling chaos in a food chain model through threshold harvesting. Fisheries and Aquaculture Journal **6**: 4.

Lotka, A. J., 1925 *Elements of Mathematical Biology*. Williams and Wilkins, Baltimore.

May, R. M., 1974 *Stability and Complexity in Model Ecosystems*. Princeton University Press.

Mobayen, S., K. V. Christos, S. Kaçar, U. Çavuşoğlu, and B. Vaseghi, 2018 A chaotic system with infinite equilibria on an exponential curve and its engineering application. International Journal of Bifurcation and Chaos **28**: 1850112.
Mobayen, S., J. Ma, G. Pujol-Vazquez, L. Acho, and Q. Zhu, 2019 Adaptive finite-time stabilization of chaotic flow with a single unstable node using nonlinear sliding mode. Iranian Journal of Science and Technology, Transactions of Electrical Engineering **43**: 339–347.
Ott, E., C. Grebogi, and J. A. Yorke, 1990 Controlling chaos. Physical Review Letters **64**: 1196–1199.
Paine, R. T., 1966 Food web complexity and species diversity. The American Naturalist **100**: 65–75.
Panday, P., N. Pal, S. Samanta, and J. Chattopadhyay, 2018 Stability and bifurcation analysis of a three-species food chain model with fear. International Journal of Bifurcation and Chaos **28**: 1850001.
Parshad, R. D., E. Quansah, U. R. K. Black, S. K. Tiwari, and N. Kumari, 2016 Long time dynamics of a three-species food chain model with allee effect in the top predator. Computers and Mathematics with Applications **71**: 503–528.
Pyragas, K., 1992 Continuous control of chaos by self-controlling feedback. Physics Letters A **170**: 421–428.
Pyragas, K., 1995 Control of chaos via extended delay feedback. Physics Letters A **206**: 323–330.
Romeiras, F. J., C. Grebogi, E. Ott, and W. P. Dayawansa, 1992 Controlling chaotic dynamical systems. Physica D **58**: 165–192.
Selvam, A. G. M., S. B. Jacob, and R. Dhineshbabu, 2020 Nonlinear dynamics in population systems. Journal of Physics: Conference Series **1543**: 012010.
Sen, M., M. Banerjee, and A. Morozov, 2012 Bifurcation analysis of a ratio-dependent prey-predator model with the allee effect. Ecological Complexity **11**: 12–27.
Skalski, G. T. and J. F. Gilliam, 2001 Functional responses with predator interference: Ecological implications. Ecology **82**: 1200–1206.
Strogatz, S. H., 2015 *Nonlinear Dynamics and Chaos*. Westview Press, second edition.
Vaseghi, B., S. Mobayen, S. S. Hashemi, and A. Fekih, 2020 Fast reaching finite time synchronization for chaotic systems with application in medical image encryption. IEEE Access **9**: 25911–25925.
Vaseghi, B., M. A. Pourmina, and S. Mobayen, 2017 Finite-time chaos synchronization and its application in wireless sensor networks. Transactions of the Institute of Measurement and Control **40**: 3788–3799.
Volterra, V., 1962 *Opere Matematiche: Memorie e Note*. Accademia Nazionale dei Lincei, Rome.
Zhang, C. H., X. P. Yan, and G. H. Cui, 2010 Hopf bifurcations in a predator-prey system with a discrete delay and a distributed delay. Nonlinear Analysis: Real World Applications **11**: 4141–4153.
Znegui, W., H. Gritli, and S. Belghith, 2020a Poincaré map for passive dynamic walking of the compass-gait biped model. Chaos, Solitons & Fractals **130**: 109436.
Znegui, W., H. Gritli, and S. Belghith, 2020b Stabilization of compass-gait biped robot by controlled poincaré map. Nonlinear Dynamics **101**: 1061–1091.
Znegui, W., H. Gritli, and S. Belghith, 2021 A new poincaré map for investigating compass-gait biped robot. Applied Mathematical Modelling **94**: 534–557.

***How to cite this article:*** Abbas, A., and Khaliq, A. Dynamics, Stability, and Bifurcation in Discrete-Time Predator-Prey Model. *ADBA Computer Science*, 3(1), 26-36, 2026.



ADBA Computer Science
Cutting-Edge Scientific Solutions

e-ISSN: 3023-8544
RESEARCH ARTICLE
Vol.3 / No.1 / 2026 / pp.37-50
https://doi.org/10.69882/adba.cs.2026016

# Stochastic Weather Simulation of Türkiye's Geographical Regions: A Monte Carlo Framework

**Nihan Hüsna Kılıç** *,[1] **and Abdullah Sevin** α,[2]

*Institute of Science, Data Science and Artificial Intelligence Program, Sakarya University, Sakarya, Türkiye, αFaculty of Computer and Information Sciences, Computer Engineering, Sakarya University, Sakarya, Türkiye.

**ABSTRACT** Understanding and predicting regional climatic variations is crucial for agricultural planning, water resource management, and disaster mitigation in the face of increasing global climate instability. This study presents a comprehensive stochastic simulation framework designed to model the diverse meteorological profiles of Türkiye's seven geographical regions. The developed model utilizes a randomized structure that learns monthly parameters from historical daily datasets spanning the 2010–2024 period and generates daily temperature and precipitation scenarios for 2025. Methodologically, a Normal Distribution was employed for temperature modeling, while a first-order Markov Chain was utilized to determine the occurrence of precipitation. To account for the characteristically positive and right-skewed nature of rainfall intensity on wet days, the Gamma Distribution was preferred a standard approach in current literature for modeling daily precipitation amounts. To capture the inherent stochasticity of atmospheric processes, a Monte Carlo simulation approach was implemented. Each representative city was initially simulated with 300 iterations to ensure statistical robustness. To rigorously assess predictive accuracy, validation for 2024 was performed using a Monte Carlo simulation with 400 independent runs. The findings demonstrate that the model effectively captures regional climatic trends and provides a reliable synthetic dataset for environmental planning. Validation metrics indicate strong agreement between modeled and observed climatic behavior, with historical monthly temperatures, precipitation totals, and wet-day probabilities consistently falling within the simulated 95% confidence intervals. These results suggest that integrated stochastic models offer a high-fidelity and computationally efficient alternative to complex numerical weather predictors for regional climate assessment.

**KEYWORDS**
Monte carlo simulation
Stochastic weather modeling
Precipitation simulation
Temperature modeling
Markov chain

## INTRODUCTION

The accurate representation and prediction of climatic variables are fundamental components of environmental modeling, influencing critical sectors such as agriculture, energy management, and urban planning. Due to the complex and chaotic nature of the atmosphere, traditional deterministic approaches often fall short of capturing the full spectrum of climatic variability. In this context, modeling and simulation approaches leverage statistical behaviors learned from historical data to generate probabilistic forecasts (Palmer *et al.* 2005).

When historical weather patterns are examined, it is observed that even under identical initial conditions, varying outcomes can emerge due to inherent atmospheric uncertainty. Therefore, relying on a single deterministic forecast is insufficient for realistic modeling (Palmer 2000). To address this, multiple potential scenarios must be simulated to measure precision and quantify uncertainty. This study aims to stochastically model the temperature and precipitation processes of seven representative cities from Türkiye's diverse climatic regions. The results are reported in terms of monthly mean temperature, the probability of precipitation, and mean total precipitation with a 95% confidence interval, aiming to achieve high-fidelity estimations. To interpret atmospheric uncertainty, the Monte Carlo simulation method was employed, allowing the same model to be executed numerous times to observe the distribution of results. To ensure stable confidence intervals while maintaining computational efficiency, 300 iterations were performed for each city.

In research focusing on stochastic weather generation, various statistical approaches are utilized to represent the random nature and temporal continuity of meteorological variables on a daily scale. Within this framework, the occurrence of precipitation is predominantly modeled using Markov chains, while the intensity of rainfall on wet days is characterized by the Gamma distribution which is widely preferred in modeling daily rainfall amounts. This methodology serves as a fundamental building block for widely recognized models such as WGEN and LARS-WG (Semenov *et al.* 1998; Wilks 1999b,a). For continuous variables like temperature and solar radiation, methods based on the Normal distribution assumption are generally preferred. Studies by Wilks have demonstrated that Monte Carlo-based simulations provide an effective framework for producing these variables in a consistent and integrated manner (Wilks 1998). Subsequent research has further investigated the extent to which these stochastic models

**Manuscript received:** 12 September 2025,
**Revised:** 18 December 2025,
**Accepted:** 20 December 2025.

[1]nihanhusnakilic@gmail.com
[2]asevin@sakarya.edu.tr (**Corresponding author**)

reflect both climate variability and extreme precipitation events across different scales. Collectively, Markov chain and probability distribution-based Monte Carlo approaches are considered established and reliable methods in the literature for generating realistic daily weather data.

The development and application of stochastic weather generators (SWGs) have been extensively documented in the literature as effective tools for bridging the gap between coarse climate data and local-scale applications. (Schuol and Abbaspour 2007) emphasized the utility of using monthly weather statistics to generate daily data, demonstrating that such approaches can provide robust inputs for hydrological models like SWAT (Soil and Water Assessment Tool). Similarly, (Kilsby *et al.* 2007)] highlighted the importance of daily weather generators in climate change studies, providing a framework. Recent advancements in the field have shifted towards more sophisticated probabilistic structures.

(Ailliot *et al.* 2015) provided a comprehensive overview of weather-type models, categorizing the evolution of SWGs and their ability to capture different atmospheric states. More recently, (Guan *et al.* 2025) evaluated the performance of regional stochastic generators, specifically focusing on their ability to reproduce heavy-precipitation events across various spatial and temporal scales. Their findings reinforce the reliability of stochastic approaches in representing extreme meteorological phenomena, a critical requirement for modern hazard and risk assessments. In parallel, recent studies have expanded stochastic weather generator capabilities by improving precipitation representation, uncertainty characterization, applicability to climate-risk assessments and simulation of extreme weather events (Najibi *et al.* 2024; Woodson *et al.* 2025; Nandan *et al.* 2024; Semenov 2008). This body of research provides a solid foundation for the methodology adopted in this study, which integrates established Markovian and distributional techniques to model the regional climate variability of Türkiye.

For this study, daily temperature and precipitation data were retrieved via the Open-Meteo Archive API (Zippenfenig 2023). Using the specific latitude and longitude coordinates for each city, daily mean temperature and total precipitation values were downloaded for the period between January 1, 2010, and December 31, 2024. Seven cities were selected to represent all geographical regions of Türkiye, enabling a comparative analysis of different climate types within a unified modeling framework. Daily observation data for the 2010–2024 period were retrieved for all representative cities to serve as the foundation for parameter estimation and simulation. To ensure physical consistency and exclude stochastic noise from trace amounts (e.g., light drizzle), a daily precipitation threshold of 1.0 mm was established.

To ensure that the stochastic projections are both robust and scientifically defensible, the modeling process is structured into two distinct stages: Validation and Future Scenario Generation. In the first stage, the model's performance is evaluated against independent historical observations from the year 2024, which were withheld during the initial parameter fitting phase. This assessment confirms the framework's ability to accurately reproduce regional climatic statistics. Following successful validation, stochastic simulations are then conducted for the future year of 2025. This dual-stage approach ensures that the projected scenarios are grounded in demonstrated statistical performance rather than simple extrapolation, providing a more reliable basis for regional climate assessment.

## METHODOLOGY

### Stochastic Framework and Simulation Logic

Atmospheric processes are inherently stochastic, and a single simulation output is insufficient to represent the full spectrum of possible climatic behaviors. Consequently, Monte Carlo-based approaches are widely employed in weather generation literature. (Wilks 1999a) demonstrated that multiple simulations are critical for uncertainty analysis in the simultaneous generation of variables such as daily temperature and precipitation. Similarly, (Knutti *et al.* 2008) emphasized that confidence intervals in climate projections can only be meaningfully interpreted through multi-scenario ensembles. In this study, temperature was modeled using a monthly Normal Distribution, precipitation occurrence via a first-order Markov Chain, and precipitation intensity—conditional on a wet day—via a Gamma Distribution. The modeling framework is based on the following assumptions:

- Daily temperature acts as a continuous random variable with a nearly symmetric distribution.
- Precipitation occurrence exhibits temporal dependency between successive days.
- Precipitation intensity is a continuous, non-negative, and right-skewed variable.

### Mathematical Modeling of Variables

Monthly mean and standard deviation values were used to sample daily temperatures from a Normal Distribution. For precipitation, since wet and dry states are not statistically independent, a first-order Markov Chain was utilized. Because wet and dry days tend to cluster in time, precipitation occurrence is commonly modeled with a two-state, first-order Markov chain, where the probability of a wet day depends on the previous day's state (Wilks 1999b,a). The process is defined as a binary state (W=1 for wet, W=0 for dry), and transition probabilities were calculated. Monthly transition probabilities were estimated directly from observed wet/dry sequences using empirical frequencies. A "wet day" was defined as having a total daily precipitation at least 1.0 mm. This threshold was selected to exclude trace events such as drizzle, which would otherwise alter the probability distribution and diminish the model's realism.

Since daily precipitation amounts are strictly positive and exhibit pronounced right-skewness, the Gamma Distribution was employed to represent asymmetric rainfall behavior. During model development, alternative formulations such as the Generalized Extreme Value (GEV) for extreme event analysis or Skew-Normal distributions for asymmetric temperature behavior were considered. However, as the primary objective of this study is to represent regional climatic variability at a monthly scale rather than focusing on rare extremes, the Gamma and Normal distributions were deemed sufficient and computationally efficient, in line with recent literature (Najibi *et al.* 2024; Woodson *et al.* 2025; Nandan *et al.* 2024).

The simulation framework follows a two-phase process: (1) Parameter Learning, where historical data are used to estimate transition probabilities and distribution parameters, and (2) Stochastic Generation, where daily variables are synthesized based on the learned states. To ensure statistical stability, the year 2025 was simulated using a 400-run Monte Carlo ensemble for each city. All simulations utilized a fixed random seed to ensure reproducibility, with calendar structures (month lengths) corresponding to the year 2024.

The year 2025 was simulated using a 300-run Monte Carlo ensemble for each city to characterize potential future variability. In the current implementation, month lengths were based on the 2024 calendar structure; this choice defines the temporal framework and does not affect the learned stochastic parameters. This ensemble approach aims to stabilize uncertainty intervals and provide a comprehensive distribution of possible outcomes. To ensure reproducibility, all simulations were generated using a fixed random seed (RNG_SEED=42). Furthermore, the model's predictive performance was rigorously tested using an extended 400-run validation ensemble for the year 2024.

#### Mathematical Framework and Operational Principles

In this study, daily weather processes are modeled using monthly parameters derived from historical observations. The model represents temperature and precipitation through separate yet internally consistent stochastic components.

- Daily Average Temperature Generation:

Daily mean temperature is generated using a Normal distribution based on monthly parameters. For a given month m, the temperature process is defined as 1:

$$T_t \sim N\left(\mu_m, \sigma_m^2\right) \tag{1}$$

where $\mu_m$ and $\sigma_m^2$ represent the mean temperature and standard deviation for month m, respectively, estimated from observation data between 2010 and 2024. The Normal distribution was selected due to the approximately symmetric distribution of temperature at the monthly scale.

- Precipitation Occurrence (First-Order Markov Chain)

Due to the temporal persistence of wet weather, the state of precipitation is modeled using a first-order Markov Chain. The precipitation state is defined as a binary random variable: $W_t \in \{0, 1\}$ where $W_t$=1 denotes a wet day and $W_t$=0 denotes a dry day. The probability of a wet day depends on the state of the previous day shown in 2 and 3:

$$P\left(W_t = 1 \mid W_{t-1} = 1\right) = p_{11,m} \tag{2}$$

$$P\left(W_t = 1 \mid W_{t-1} = 0\right) = p_{01,m} \tag{3}$$

Here, $p_{11,m}$ and $p_{01,m}$ represent the conditional transition probabilities for month m. These probabilities were estimated directly from empirical wet/dry transition frequencies within the historical record. In instances or specific months where precipitation events are extremely rare, transition probabilities may approach boundary values (0 or 1). This behavior is maintained within the model as it accurately reflects the observed scarcity of meteorological transitions in the specific regional climate history, ensuring that the synthetic series remains grounded in empirical reality.

- Daily Precipitation Amount

The daily total precipitation amount is modeled conditionally upon the occurrence of a wet day. For a day t where $W_t$=1, the amount $P_t$ follows a Gamma distribution as 4:

$$P_t \mid (W_t = 1) \sim \Gamma\left(k_m, \theta_m\right) \tag{4}$$

If $W_t = 0$, then $P_t$=0. The Gamma distribution was chosen to represent the characteristic right-skewed nature of rainfall intensity, where values are strictly positive and extreme amounts occur with low frequency. Since daily precipitation amounts are strictly positive and exhibit pronounced right-skewness, particularly due to intermittent heavy rainfall events, the Gamma distribution was employed to represent rainfall intensity on wet days. Although alternative distributions (e.g., mixed exponential or extreme value formulations) may outperform the Gamma distribution in specific settings, Gamma remains a common and practical choice in stochastic weather generators, particularly when the focus is on monthly-scale variability rather than extremes (Semenov *et al.* 1998; Wilks 1999b). The shape parameter ($k_m$) and scale parameter ($\theta_m$) were estimated from wet-day observations within the corresponding month.

- Uncertainty Evaluation (Monte Carlo Simulation)

To evaluate uncertainty, the model was iterated 300 times for each representative city. All realizations were generated as stochastic draws from a single seeded random-number generator stream, ensuring reproducibility while yielding distinct ensemble members. The 95% confidence intervals for monthly metrics were calculated using the quantiles of the resulting Monte Carlo sample distributions in 5:

$$CI_{95\%} = [Q_{0.025}, Q_{0.975}] \tag{5}$$

This approach allows for a quantitative assessment of not only the average behavior but also the inherent variability and uncertainty in the synthetic weather data. Recent studies highlight that ensemble-based confidence bands provide an interpretable way to communicate uncertainty in climate-sensitive applications (Woodson *et al.* 2025; Nandan *et al.* 2024). For the 2025 simulation experiments, 300 Monte Carlo realizations were employed. This ensemble size was selected to balance statistical stability in monthly means and 95% confidence intervals with computational efficiency. Overall, the literature indicates that Markov-chain-based occurrence models combined with parametric distributions—such as the Gamma and Normal distributions used in this study—can effectively reproduce realistic mean behavior and variability at daily to monthly scales (Ailliot *et al.* 2015).

## RESULTS AND DISCUSSION

Stochastic weather generators based on Markov chains and parametric probability distributions have been proven to realistically reproduce both mean climate behavior and variability at daily and monthly scales (Ailliot *et al.* 2015). Furthermore, recent studies highlight that Monte Carlo ensembles and probabilistic summaries significantly improve interpretability when comparing simulated outcomes against observed climate indicators (Najibi *et al.* 2024; Woodson *et al.* 2025; Nandan *et al.* 2024). Building on this established framework, this section presents the outputs of the stochastic weather simulation for the year 2025 across seven representative cities of Türkiye.

The findings are categorized into monthly temperature trends, precipitation occurrences, and total rainfall intensity. By utilizing the 300-run Monte Carlo ensemble, the results provide not only the expected mean values but also the probabilistic boundaries of potential climatic shifts. The following sub-sections evaluate the model's performance in capturing regional climatic characteristics ranging from the humid subtropical profile of the Black Sea to the semi-arid conditions of Central and Southeastern Anatolia through comparative tables and visual distribution plots.

The temperature trends, as shown in Figure 1, indicate that the lowest mean temperatures occur during January and February. A steady upward trend is observed during the summer months,

followed by a gradual cooling phase starting from autumn. Similar to the precipitation patterns, the 95% confidence interval for temperature is broader in the winter and more constrained in the summer. This finding suggests that winter temperatures are subject to greater stochastic variability, while summer heatwaves or average temperatures follow a more predictable trajectory. These results are in full alignment with the transitional climate characteristics of the Marmara Region.

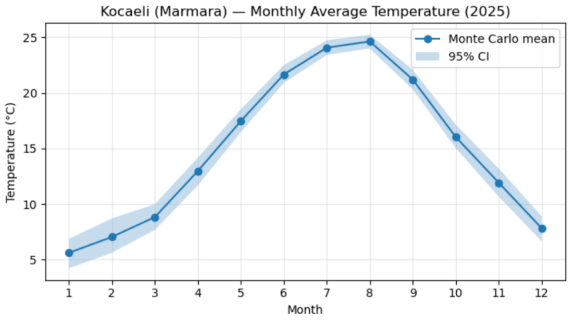



**Figure 1** Monthly Average Temperature of Kocaeli

The simulation results for the Marmara region, represented by Kocaeli, demonstrate high seasonal consistency with historical climatic patterns. As illustrated in Figure 2, the probability of precipitation reaches its peak during the winter months and significantly declines throughout the summer period. The 95% confidence interval is notably wider during the winter, suggesting that precipitation events in this season exhibit higher irregularity and variance. Conversely, the narrow confidence interval observed in the summer months indicates that dry conditions are more stable and frequent during this period.

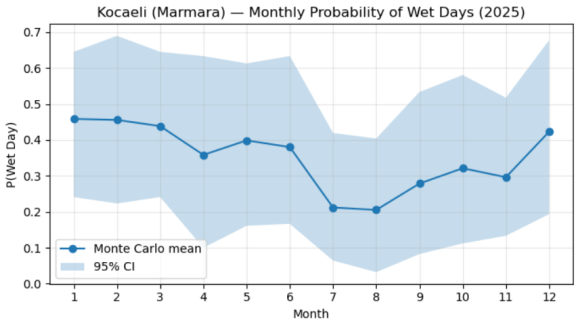



**Figure 2** Monthly Probability of Wet Days of Kocaeli

Furthermore, the monthly precipitation intensity values presented in Table 1 confirm that the highest total rainfall amounts are concentrated in the autumn and winter seasons. The significant decrease in total precipitation during the summer months highlights the model's ability to accurately reflect the regional dry season.

Regarding temperature trends shown in Figure 3, the simulation results are highly consistent with the regional climate. The 95% confidence interval for temperature remains remarkably stable and narrow throughout the year, indicating low inter-annual temperature variability for this province. This suggests that daily mean temperatures in Izmir follow a more predictable seasonal trajectory compared to other regions.

The simulation results for Izmir, as the representative city of the Aegean Region, accurately reflect the transition between seasons. As illustrated in Figure 4, the probability of precipitation decreases

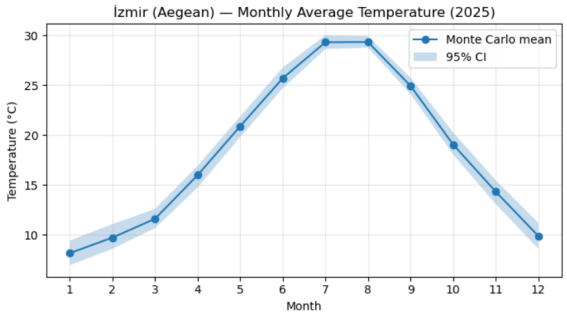



**Figure 3** Monthly Average Temperature of Izmir

gradually from winter to spring, shows a slight increase during the mid-spring period, and reaches its minimum during the summer months before trending upward in autumn. Notably, the precipitation probability in July and August is near zero, suggesting that these months may remain entirely dry under typical conditions.

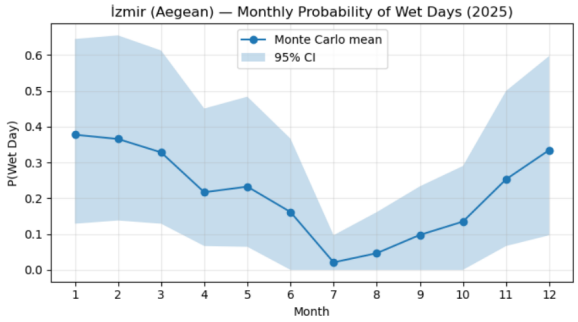



**Figure 4** Monthly Probability of Wet Days of Izmir

A comparative analysis of monthly total precipitation and the probability of wet days (Table 2) reveals a high degree of correlation. However, the widening of confidence intervals specifically during the winter months indicates that uncertainty in precipitation intensity is significantly higher than that of temperature. Overall, the stochastic outputs for Izmir are in full agreement with the characteristic Mediterranean-influenced Aegean climate, capturing both the summer aridity and the variability of winter rainfall.

The simulation results for Antalya, as shown in Figure 5, are in high alignment with the classical Mediterranean climate profile. In months where the 95% confidence interval is narrow, a stable thermal regime prevails. Conversely, in months where the interval relatively widens, it can be inferred that meteorological uncertainty and inter-annual variability are higher. Overall, the model successfully captures the "hot-summer Mediterranean" (Csa) characteristics, particularly highlighting the high variance in winter precipitation against the predictable aridity of the summer.

As illustrated in Figure 6, the probability of wet days in Antalya significantly increases during the winter months while exhibiting a sharp decline during the summer. The narrow confidence intervals observed during the summer months indicate that the absence of precipitation is statistically stable and consistent during this period. An analysis of monthly total precipitation amounts (Table 3) reveals that rainfall is concentrated in the winter season, both in terms of frequency and intensity.

The simulation results presented in Figure 7 are highly consistent with the typical continental climate of Central Anatolia,

**Table 1** Monthly statistical summary of simulated temperature and precipitation parameters of Kocaeli

| Month | $T_{mean}$ | $T_{low}$ | $T_{high}$ | $W_{prob}$ | $W_{low}$ | $W_{high}$ | $P_{mean}$ | $P_{low}$ | $P_{high}$ |
|---|---|---|---|---|---|---|---|---|---|
| 01 | 5.61 | 4.21 | 6.86 | 0.46 | 0.24 | 0.65 | 96.24 | 37.27 | 162.59 |
| 02 | 7.04 | 5.62 | 8.71 | 0.46 | 0.22 | 0.69 | 84.35 | 29.15 | 153.88 |
| 03 | 8.83 | 7.67 | 9.98 | 0.44 | 0.24 | 0.65 | 83.75 | 39.21 | 150.66 |
| 04 | 12.97 | 11.67 | 14.17 | 0.36 | 0.10 | 0.63 | 67.47 | 16.34 | 130.52 |
| 05 | 17.47 | 16.48 | 18.53 | 0.40 | 0.16 | 0.61 | 74.33 | 24.58 | 132.77 |
| 06 | 21.66 | 20.91 | 22.49 | 0.38 | 0.17 | 0.63 | 79.54 | 18.33 | 149.30 |
| 07 | 24.07 | 23.42 | 24.73 | 0.21 | 0.06 | 0.42 | 37.36 | 3.35 | 97.18 |
| 08 | 24.63 | 24.01 | 25.23 | 0.21 | 0.03 | 0.40 | 33.17 | 0.70 | 89.40 |
| 09 | 21.20 | 20.32 | 22.08 | 0.28 | 0.08 | 0.53 | 54.12 | 5.52 | 119.56 |
| 10 | 16.04 | 15.05 | 17.15 | 0.32 | 0.11 | 0.58 | 84.55 | 15.07 | 191.06 |
| 11 | 11.92 | 10.67 | 13.20 | 0.30 | 0.13 | 0.52 | 69.32 | 17.75 | 136.07 |
| 12 | 7.82 | 6.66 | 8.90 | 0.42 | 0.19 | 0.68 | 100.94 | 34.84 | 181.49 |

**Table 2** Monthly statistical summary of simulated temperature and precipitation parameters of Izmir

| Month | $T_{mean}$ | $T_{low}$ | $T_{high}$ | $W_{prob}$ | $W_{low}$ | $W_{high}$ | $P_{mean}$ | $P_{low}$ | $P_{high}$ |
|---|---|---|---|---|---|---|---|---|---|
| 01 | 8.18 | 6.97 | 9.44 | 0.38 | 0.13 | 0.65 | 130.35 | 29.17 | 251.79 |
| 02 | 9.73 | 8.61 | 11.08 | 0.37 | 0.14 | 0.66 | 95.37 | 22.50 | 192.31 |
| 03 | 11.62 | 10.69 | 12.60 | 0.33 | 0.13 | 0.61 | 79.79 | 12.71 | 167.38 |
| 04 | 15.99 | 14.75 | 16.92 | 0.22 | 0.07 | 0.45 | 54.53 | 4.96 | 129.37 |
| 05 | 20.89 | 19.81 | 21.86 | 0.23 | 0.06 | 0.48 | 45.69 | 6.79 | 107.60 |
| 06 | 25.70 | 24.69 | 26.77 | 0.16 | 0.00 | 0.37 | 27.47 | 0.00 | 71.77 |
| 07 | 29.30 | 28.62 | 29.99 | 0.02 | 0.00 | 0.10 | 2.00 | 0.00 | 10.66 |
| 08 | 29.32 | 28.77 | 29.91 | 0.05 | 0.00 | 0.16 | 3.98 | 0.00 | 15.49 |
| 09 | 24.93 | 24.11 | 25.74 | 0.10 | 0.00 | 0.23 | 18.13 | 0.00 | 62.27 |
| 10 | 19.06 | 18.06 | 20.28 | 0.13 | 0.00 | 0.29 | 55.01 | 0.00 | 169.76 |
| 11 | 14.34 | 13.09 | 15.48 | 0.25 | 0.07 | 0.50 | 80.87 | 10.02 | 192.93 |
| 12 | 9.89 | 8.63 | 11.21 | 0.33 | 0.10 | 0.60 | 111.20 | 16.15 | 240.03 |

characterized by hot, arid summers and cold, unstable winters. The model accurately captures the seasonal fluctuations in both temperature and rainfall, providing a realistic representation of the region's atmospheric dynamics.

As illustrated in Figure 8, the probability of wet days in Ankara is notably higher during the spring months compared to the summer, with a further increasing trend observed throughout the autumn. The widening of the confidence intervals during these transitional periods suggests that the precipitation regime (Table 4) in Ankara exhibits significant variability during seasonal shifts.

According to the temperature trends shown in Figure 9, the relatively narrow confidence intervals indicate a stable thermal regime. The data suggests that Trabzon experiences a temperate climate where extreme temperature fluctuations are mitigated. This stability reflects the moderating influence of the sea, which prevents drastic seasonal shifts and maintains a consistent temperature trajectory as captured by the simulation model.

As illustrated in Figure 10, the probability of wet days in Trabzon remains high throughout the year, with this trend persisting even during the summer months. This result is highly consistent with the characteristic year-round precipitation regime of the Black Sea region.

The breadth of the confidence intervals across the simulated period suggests that while rainfall is frequent, its daily intensity and occurrence exhibit significant stochastic variability in the representative city. The comprehensive statistical breakdown of these

**Table 3** Monthly statistical summary of simulated temperature and precipitation parameters of Antalya

| Month | $T_{mean}$ | $T_{low}$ | $T_{high}$ | $W_{prob}$ | $W_{low}$ | $W_{high}$ | $P_{mean}$ | $P_{low}$ | $P_{high}$ |
|---|---|---|---|---|---|---|---|---|---|
| 01 | 9.43 | 8.55 | 10.33 | 0.42 | 0.19 | 0.68 | 242.21 | 74.93 | 465.64 |
| 02 | 10.43 | 9.54 | 11.26 | 0.36 | 0.14 | 0.66 | 135.95 | 23.24 | 300.43 |
| 03 | 12.39 | 11.59 | 13.16 | 0.29 | 0.10 | 0.52 | 96.88 | 15.63 | 220.63 |
| 04 | 16.05 | 15.22 | 16.87 | 0.23 | 0.07 | 0.43 | 54.21 | 7.56 | 120.09 |
| 05 | 20.38 | 19.51 | 21.26 | 0.24 | 0.03 | 0.45 | 59.66 | 3.80 | 165.13 |
| 06 | 25.21 | 23.97 | 26.46 | 0.13 | 0.00 | 0.30 | 17.72 | 0.00 | 49.94 |
| 07 | 29.45 | 28.76 | 30.21 | 0.04 | 0.00 | 0.16 | 5.13 | 0.00 | 30.62 |
| 08 | 29.26 | 28.69 | 29.96 | 0.05 | 0.00 | 0.16 | 5.44 | 0.00 | 22.72 |
| 09 | 25.81 | 25.07 | 26.58 | 0.08 | 0.00 | 0.23 | 22.58 | 0.00 | 85.50 |
| 10 | 20.31 | 19.50 | 21.21 | 0.19 | 0.03 | 0.39 | 105.56 | 0.51 | 277.51 |
| 11 | 15.25 | 14.48 | 15.98 | 0.24 | 0.03 | 0.50 | 103.83 | 1.59 | 254.29 |
| 12 | 11.12 | 10.17 | 11.95 | 0.40 | 0.14 | 0.68 | 207.76 | 41.94 | 418.76 |

**Table 4** Monthly statistical summary of simulated temperature and precipitation parameters of Ankara

| Month | $T_{mean}$ | $T_{low}$ | $T_{high}$ | $W_{prob}$ | $W_{low}$ | $W_{high}$ | $P_{mean}$ | $P_{low}$ | $P_{high}$ |
|---|---|---|---|---|---|---|---|---|---|
| 01 | 1.11 | -0.64 | 2.80 | 0.32 | 0.13 | 0.52 | 56.52 | 18.71 | 110.80 |
| 02 | 3.50 | 1.86 | 5.12 | 0.28 | 0.07 | 0.55 | 39.61 | 4.64 | 82.09 |
| 03 | 6.37 | 4.96 | 7.85 | 0.33 | 0.11 | 0.58 | 55.79 | 14.02 | 108.68 |
| 04 | 11.91 | 10.53 | 13.26 | 0.29 | 0.08 | 0.50 | 35.99 | 6.37 | 72.16 |
| 05 | 16.49 | 15.14 | 17.53 | 0.34 | 0.14 | 0.60 | 53.81 | 16.59 | 108.31 |
| 06 | 20.52 | 19.56 | 21.56 | 0.30 | 0.07 | 0.57 | 43.97 | 5.43 | 88.99 |
| 07 | 24.49 | 23.57 | 25.35 | 0.06 | 0.00 | 0.16 | 6.42 | 0.00 | 22.83 |
| 08 | 25.00 | 24.18 | 25.86 | 0.08 | 0.00 | 0.23 | 7.62 | 0.00 | 24.84 |
| 09 | 20.61 | 19.42 | 21.93 | 0.07 | 0.00 | 0.20 | 12.66 | 0.00 | 61.68 |
| 10 | 13.97 | 12.63 | 15.27 | 0.15 | 0.00 | 0.37 | 25.83 | 0.00 | 65.45 |
| 11 | 8.05 | 6.81 | 9.36 | 0.16 | 0.00 | 0.37 | 33.38 | 0.00 | 84.86 |
| 12 | 3.38 | 2.18 | 4.49 | 0.28 | 0.06 | 0.52 | 49.36 | 6.94 | 106.79 |

parameters is provided in Table 5.

As illustrated in Figure 11 and Figure 12, the simulation results for Erzurum are highly consistent with the severe continental climate of Eastern Anatolia. During the winter months, mean temperatures remain significantly below freezing (negative Celsius values). The relatively wide confidence intervals observed in this period suggest that the region is subject to extreme cold spells and high thermal variability.

Regarding precipitation patterns in Table 6, the probability of wet days reaches its peak during the spring months and exhibits a notable decline during the summer season. These findings reflect the regional characteristics where snowmelt and spring convective rains are dominant, followed by a relatively drier but short summer period. The model accurately captures the harsh seasonal transitions and the rigorous winter conditions unique to the high-altitude geography of Erzurum.

As illustrated in Figure 13, the simulated mean temperatures for Diyarbakır exceed 30°C during the summer months and fall below 5°C during the winter. The prevalence of a narrow confidence interval suggests that these high temperatures are observed with significant stability and predictable seasonal intensity. Furthermore, the results in Figure 14 indicate that the probability of a wet day is nearly zero during the summer season, while it shows a marked increase during the winter months. These findings and also Table 7 are in full alignment with the semi-arid climatic characteristics of Southeastern Anatolia, specifically reflecting the intense

**Table 5** Monthly statistical summary of simulated temperature and precipitation parameters of Trabzon

| Month | $T_{mean}$ | $T_{low}$ | $T_{high}$ | $W_{prob}$ | $W_{low}$ | $W_{high}$ | $P_{mean}$ | $P_{low}$ | $P_{high}$ |
|---|---|---|---|---|---|---|---|---|---|
| 01 | 6.61 | 5.33 | 7.83 | 0.43 | 0.19 | 0.69 | 102.77 | 36.61 | 176.18 |
| 02 | 7.45 | 6.20 | 8.76 | 0.45 | 0.24 | 0.69 | 78.88 | 32.24 | 128.78 |
| 03 | 8.76 | 7.30 | 10.07 | 0.52 | 0.32 | 0.73 | 114.02 | 50.38 | 185.65 |
| 04 | 12.65 | 11.39 | 13.95 | 0.46 | 0.20 | 0.70 | 105.48 | 36.01 | 185.68 |
| 05 | 16.56 | 15.54 | 17.64 | 0.51 | 0.27 | 0.74 | 131.67 | 56.32 | 219.43 |
| 06 | 20.67 | 19.86 | 21.26 | 0.54 | 0.27 | 0.80 | 117.93 | 51.86 | 200.87 |
| 07 | 22.77 | 22.23 | 23.37 | 0.52 | 0.29 | 0.74 | 106.25 | 41.71 | 193.97 |
| 08 | 23.54 | 23.01 | 24.03 | 0.56 | 0.29 | 0.81 | 117.68 | 34.08 | 238.98 |
| 09 | 20.71 | 19.88 | 21.50 | 0.54 | 0.28 | 0.77 | 155.39 | 65.74 | 268.90 |
| 10 | 16.41 | 15.39 | 17.33 | 0.51 | 0.26 | 0.74 | 180.33 | 70.63 | 305.10 |
| 11 | 12.36 | 11.19 | 13.62 | 0.38 | 0.13 | 0.65 | 107.86 | 27.00 | 199.07 |
| 12 | 8.84 | 7.51 | 10.04 | 0.35 | 0.16 | 0.61 | 84.85 | 26.09 | 172.29 |

**Table 6** Monthly statistical summary of simulated temperature and precipitation parameters of Erzurum

| Month | $T_{mean}$ | $T_{low}$ | $T_{high}$ | $W_{prob}$ | $W_{low}$ | $W_{high}$ | $P_{mean}$ | $P_{low}$ | $P_{high}$ |
|---|---|---|---|---|---|---|---|---|---|
| 01 | -5.86 | -7.31 | -4.45 | 0.29 | 0.10 | 0.50 | 42.58 | 9.73 | 81.93 |
| 02 | -4.32 | -5.67 | -2.92 | 0.28 | 0.07 | 0.52 | 34.92 | 7.61 | 70.73 |
| 03 | -0.33 | -1.64 | 0.85 | 0.43 | 0.23 | 0.65 | 77.54 | 33.11 | 130.13 |
| 04 | 5.47 | 4.29 | 6.61 | 0.51 | 0.30 | 0.77 | 81.67 | 36.19 | 135.67 |
| 05 | 10.32 | 9.21 | 11.21 | 0.64 | 0.42 | 0.84 | 111.88 | 62.88 | 167.46 |
| 06 | 15.55 | 14.60 | 16.60 | 0.40 | 0.17 | 0.63 | 65.33 | 22.36 | 118.54 |
| 07 | 18.91 | 18.07 | 19.77 | 0.27 | 0.10 | 0.48 | 43.51 | 10.02 | 82.22 |
| 08 | 19.94 | 18.99 | 20.75 | 0.16 | 0.03 | 0.32 | 20.46 | 2.28 | 48.01 |
| 09 | 15.61 | 14.47 | 16.63 | 0.17 | 0.00 | 0.37 | 22.07 | 0.00 | 53.15 |
| 10 | 8.69 | 7.58 | 9.82 | 0.27 | 0.06 | 0.53 | 39.84 | 7.36 | 85.38 |
| 11 | 2.07 | 0.83 | 3.27 | 0.23 | 0.03 | 0.53 | 36.32 | 1.97 | 86.45 |
| 12 | -3.68 | -5.22 | -2.21 | 0.23 | 0.06 | 0.50 | 36.79 | 5.03 | 87.59 |

summer aridity and the concentrated winter precipitation regime. The model effectively captures the regional contrast between the extreme heat of the dry season and the relatively mild, wet winter period.

The aggregated simulation results in Table 8 for the seven geographical regions of Türkiye are summarized in the table above, providing a comparative overview of annual average temperatures and the probability of wet days, illustrated in also Figure 15 and Figure 16. The stochastic model demonstrates high sensitivity to regional variations, effectively distinguishing between the maritime, continental, and transitional climates of the country. According to the synthesized data, Antalya (Mediterranean) and Izmir (Aegean) exhibit the highest annual average temperatures, approximately 18.8°C and 18.3°C respectively, reflecting their subtropical characteristics. In stark contrast, Erzurum (Eastern Anatolia) presents the lowest annual mean temperature at 6.9°C, consistent with its high-altitude, severe continental climate. Regarding precipitation, Trabzon (Black Sea) stands out as the most humid region with an annual total of 1397.2 mm and the highest frequency of rainy days (174 days), emphasizing its year-round wet climate. Ankara (Central Anatolia) and Diyarbakır (Southeastern Anatolia) recorded the lowest annual rainfall totals (425.2 mm and 594.4 mm respectively), highlighting the semi-arid nature of the inland steppes. The consistency in the number of rainy days across 300 iterations (ranging from 78 to 174 days depending on the region) validates the robustness of the 1st-order Markov Chain and the 1.0 mm wet threshold

**Table 7** Monthly statistical summary of simulated temperature and precipitation parameters of Diyarbakır

| Month | $T_{mean}$ | $T_{low}$ | $T_{high}$ | $W_{prob}$ | $W_{low}$ | $W_{high}$ | $P_{mean}$ | $P_{low}$ | $P_{high}$ |
|---|---|---|---|---|---|---|---|---|---|
| 01 | 3.21 | 2.16 | 4.20 | 0.38 | 0.10 | 0.65 | 94.45 | 24.63 | 183.92 |
| 02 | 5.04 | 3.86 | 6.33 | 0.35 | 0.14 | 0.62 | 72.44 | 21.38 | 145.61 |
| 03 | 8.79 | 7.72 | 9.95 | 0.44 | 0.18 | 0.65 | 106.55 | 26.81 | 202.81 |
| 04 | 14.27 | 13.26 | 15.34 | 0.35 | 0.10 | 0.63 | 74.84 | 14.81 | 155.50 |
| 05 | 19.25 | 18.04 | 20.41 | 0.31 | 0.10 | 0.53 | 57.25 | 12.18 | 118.40 |
| 06 | 26.58 | 25.48 | 27.55 | 0.06 | 0.00 | 0.17 | 4.08 | 0.00 | 12.86 |
| 07 | 31.36 | 30.74 | 32.11 | 0.00 | 0.00 | 0.03 | 0.00 | 0.00 | 0.00 |
| 08 | 31.34 | 30.77 | 31.89 | 0.00 | 0.00 | 0.03 | 0.00 | 0.00 | 0.00 |
| 09 | 26.16 | 25.30 | 27.11 | 0.03 | 0.00 | 0.15 | 4.38 | 0.00 | 30.35 |
| 10 | 18.33 | 17.21 | 19.39 | 0.15 | 0.00 | 0.35 | 38.82 | 0.00 | 111.38 |
| 11 | 10.25 | 9.11 | 11.41 | 0.21 | 0.03 | 0.47 | 54.73 | 7.28 | 135.75 |
| 12 | 4.97 | 3.91 | 6.03 | 0.31 | 0.06 | 0.57 | 79.04 | 16.78 | 164.34 |

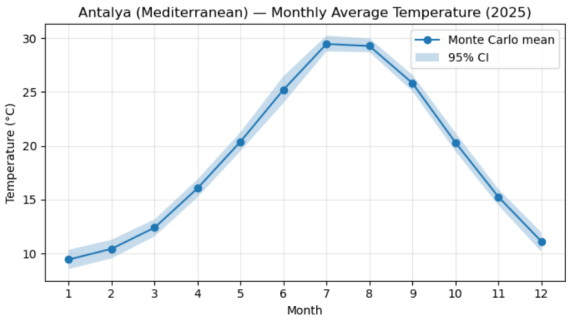



**Figure 5** Monthly Average Temperature of Antalya

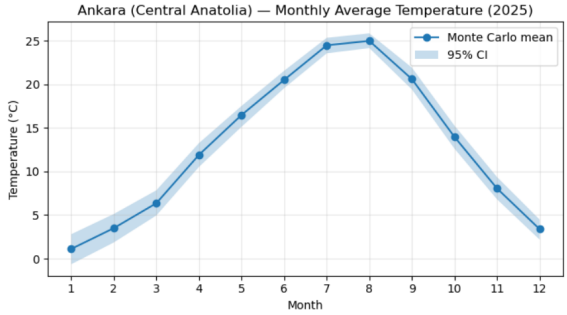



**Figure 7** Monthly Average Temperature of Ankara

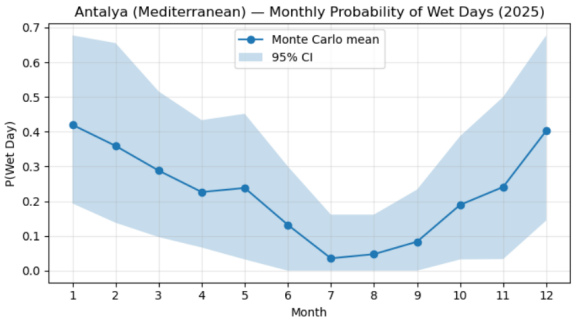



**Figure 6** Monthly Probability of Wet Days of Antalya

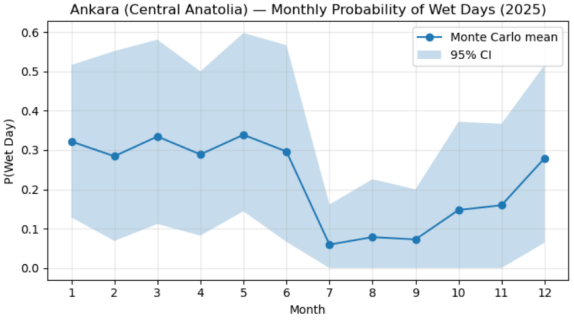



**Figure 8** Monthly Probability of Wet Days of Ankara

used in the study. These annual summaries confirm that the integrated stochastic framework can successfully replicate complex regional meteorological balances within a unified mathematical structure.

## VALIDATION RESULTS

Model validation was conducted using observed daily climate records from 2024, which were deliberately excluded from the parameter estimation stage and reserved for a strict out-of-sample evaluation. In order to assess uncertainty in a stable and reproducible manner, the stochastic framework was independently simulated for the validation year using a 400-run Monte Carlo ensemble. Although the input data are daily, validation was performed using monthly aggregated indicators, which provide a clear regional-scale comparison. Specifically, the evaluation focused on monthly average temperature, monthly total precipitation, and wet-day probability (fraction of wet days within each

**Table 8** Statistical summary of simulated temperature and precipitation parameters of Regions

| Region | City | Avg. Temp (°C) | Total Rain (mm) | Rainy Days | Runs | Threshold |
|---|---|---|---|---|---|---|
| Aegean | Izmir | 18.27 | 704.37 | 78 | 300 | 1.0 |
| Black Sea | Trabzon | 14.80 | 1403.11 | 176 | 300 | 1.0 |
| Central Anatolia | Ankara | 12.98 | 420.96 | 81 | 300 | 1.0 |
| Eastern Anatolia | Erzurum | 6.90 | 612.92 | 118 | 300 | 1.0 |
| Marmara | Kocaeli | 14.96 | 865.14 | 129 | 300 | 1.0 |
| Mediterranean | Antalya | 18.78 | 1056.93 | 81 | 300 | 1.0 |
| Southeastern Anatolia | Diyarbakır | 16.66 | 586.58 | 79 | 300 | 1.0 |

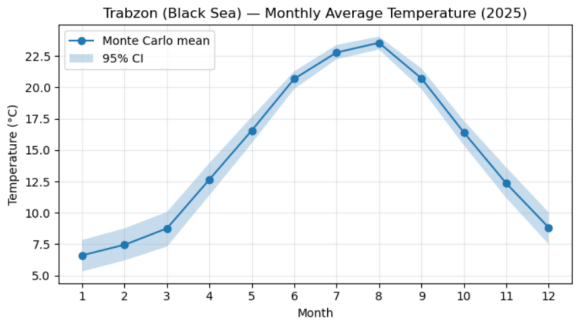



**Figure 9** Monthly Average Temperature of Trabzon

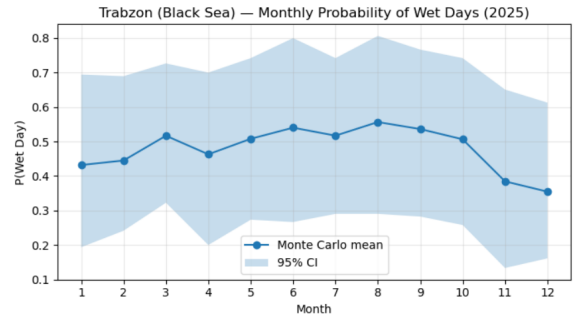



**Figure 10** Monthly Probability of Wet Days of Trabzon

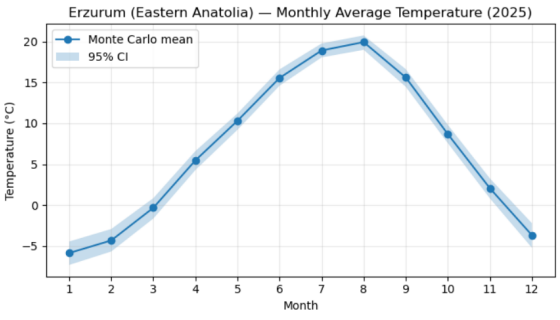



**Figure 11** Monthly Average Temperature of Erzurum

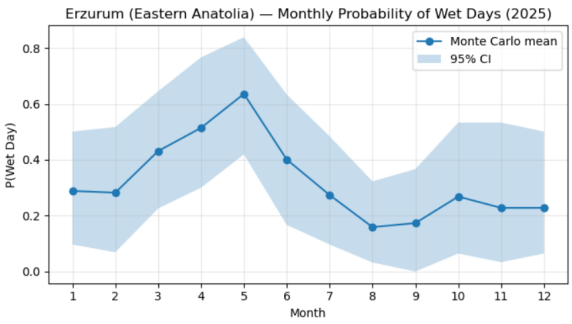



**Figure 12** Monthly Probability of Wet Days of Erzurum

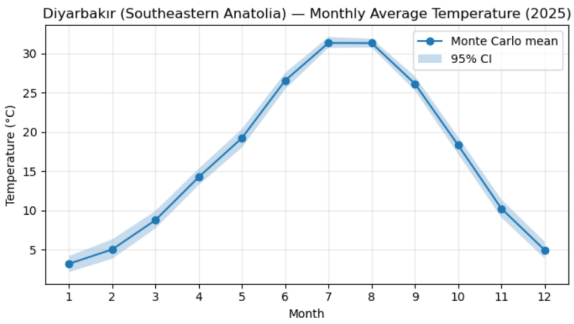



**Figure 13** Monthly Average Temperature of Diyarbakır

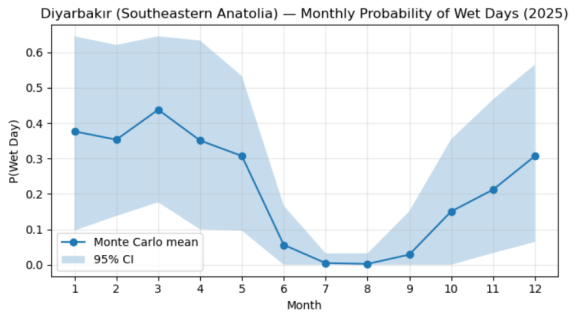



**Figure 14** Monthly Probability of Wet Days of Diyarbakır

month). For each month, the simulation results are summarized using the ensemble mean together with 95% uncertainty bounds derived from the 2.5th and 97.5th percentiles of the Monte Carlo distribution.

Figure 17 presents the validation results for Kocaeli, representing the Marmara Region. Observed monthly average temperature

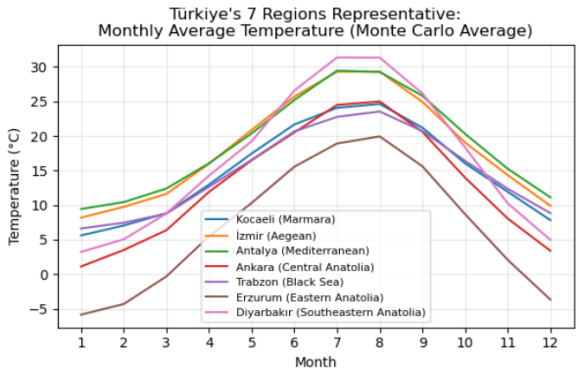



**Figure 15** Average Temperature of Regions

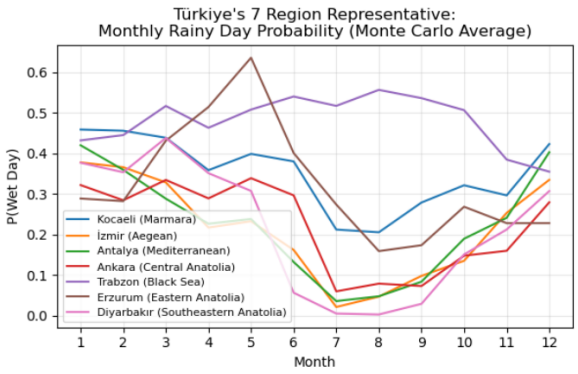



**Figure 16** Probability of Wet Days of Regions

values for 2024 closely follow the Monte Carlo ensemble mean and remain within the simulated 95% uncertainty bounds throughout the year. Deviations are generally minor, indicating that the model successfully reproduces the seasonal thermal cycle characteristic of the Marmara climate. Precipitation-related indicators show noticeably wider uncertainty ranges, particularly during winter, reflecting the irregular and event-driven nature of rainfall. Despite this variability, observed monthly total precipitation and wet-day probability remain consistently captured within the ensemble uncertainty bounds. Overall, the results confirm that the combined first-order Markov Chain and Gamma distribution framework provides a reliable representation of both precipitation frequency and intensity for the Marmara Region.

Figure 18 shows validation results for Izmir, representing the Aegean Region. Observed monthly average temperatures align closely with the ensemble mean and remain almost entirely within the 95% uncertainty bounds, which stay relatively narrow across the year. This reflects the strong seasonal predictability of temperature in the Aegean climate. In contrast, precipitation-related variables exhibit broader uncertainty intervals, particularly in winter months, due to higher stochastic variability in rainfall processes. Nevertheless, the model successfully captures the dominant regional pattern, including the pronounced summer dry period. Both observed monthly precipitation totals and wet-day probabilities remain within the simulated uncertainty ranges, confirming that the stochastic framework realistically represents the seasonal precipitation regime typical of the Aegean Region.

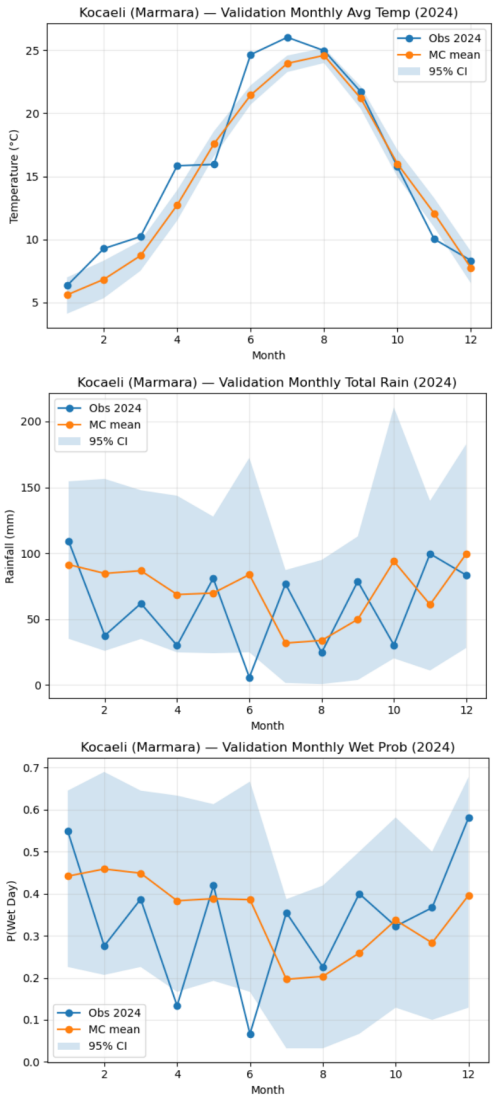



**Figure 17** Validation of monthly average temperature, total precipitation, and wet-day probability for Kocaeli (Marmara Region) in 2024

Figure 19 presents the validation results for Antalya, representing the Mediterranean Region. Observed monthly average temperature values closely track the Monte Carlo ensemble mean and consistently fall within the simulated uncertainty bounds. The confidence bands remain particularly narrow during summer, reflecting the stable thermal regime associated with persistent hot-season conditions. Precipitation-related indicators show strong seasonality, with wet-day probability and total precipitation peaking during winter and decreasing sharply during summer. Wider uncertainty bounds during winter indicate increased variability linked to episodic heavy rainfall events, whereas narrow summer uncertainty reflects statistically stable dry conditions. These results demonstrate that the proposed stochastic framework captures the characteristic winter-wet and summer-dry structure of the Mediterranean climate with strong agreement to observations.

Figure 20 illustrates validation results for Ankara, representing the Central Anatolia Region. Observed monthly mean temperatures closely follow the ensemble mean and remain within the 95% uncertainty bounds, indicating that the model accurately reproduces the continental seasonal transition between cold winters

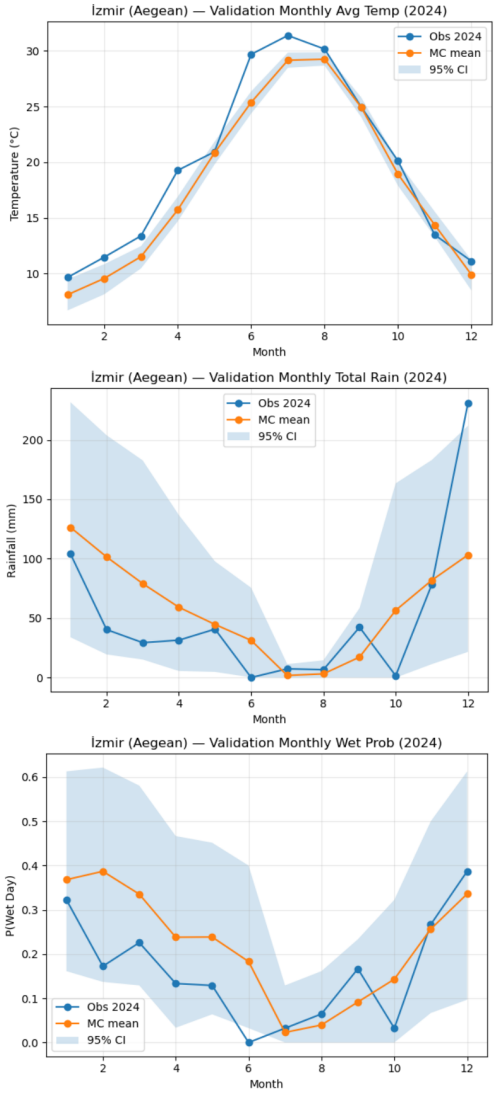



**Figure 18** Validation of monthly average temperature, total precipitation, and wet-day probability for Izmir (Aegean Region) in 2024

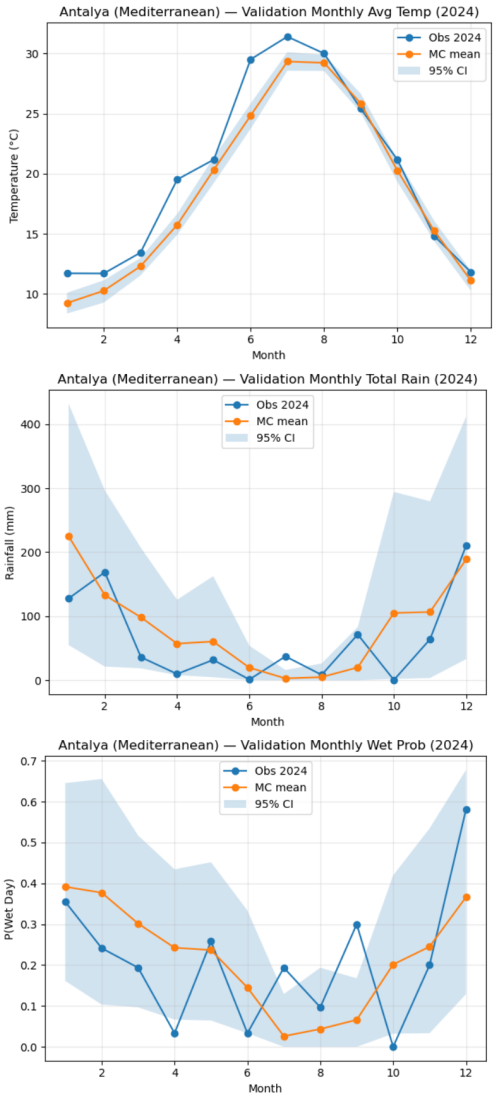



**Figure 19** Validation of monthly average temperature, total precipitation, and wet-day probability for Antalya (Mediterranean Region) in 2024

and warm summers. Precipitation-related indicators exhibit more variability during transitional seasons. Wet-day probability and precipitation totals increase during spring and early summer, followed by a marked decline in mid-summer. The widening of uncertainty bounds during spring suggests increased stochastic variability associated with convective precipitation events, while narrower summer intervals reflect a more stable dry regime. Overall, the results confirm that the framework successfully represents both the thermal cycle and the seasonal timing of precipitation typical of Central Anatolia.

Figure 21 presents validation results for Trabzon, representing the Black Sea Region. Observed monthly mean temperatures remain close to the ensemble mean and stay within the simulated uncertainty bounds, reflecting the relatively moderated thermal variability influenced by the Black Sea. Rainfall-related indicators display consistently high wet-day probabilities across all seasons, including summer months, which is a defining feature of the Black Sea climate. Although uncertainty bounds for precipitation totals and wet-day probability are wider due to variability in rainfall intensity and timing, observed values remain well within the simulated ranges. These findings indicate that the stochastic framework effectively captures the year-round humid precipitation regime characteristic of the Black Sea Region.

Figure 22 shows validation results for Erzurum, representing the Eastern Anatolia Region. Observed monthly average temperatures closely follow the ensemble mean and remain predominantly within the simulated uncertainty bounds. The model successfully reproduces the severe winter conditions, including sustained sub-zero temperatures, reflecting the high-altitude continental climate of Eastern Anatolia. Precipitation-related indicators show pronounced seasonality, with wet-day probability and precipitation totals peaking during spring and decreasing during summer. Wider uncertainty bounds during winter and spring indicate increased variability associated with harsh winter dynamics and seasonal transitions. Despite these challenges, observed values remain within the modeled uncertainty ranges, demonstrating that the framework captures the extreme seasonal contrasts typical of the Eastern Anatolia climate.

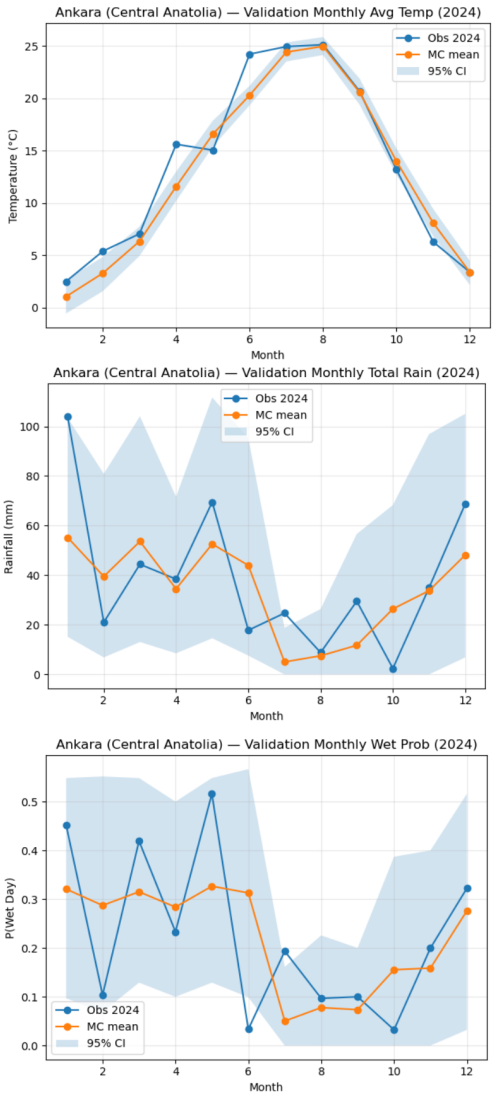



**Figure 20** Validation of monthly average temperature, total precipitation, and wet-day probability for Ankara (Central Anatolia Region) in 2024

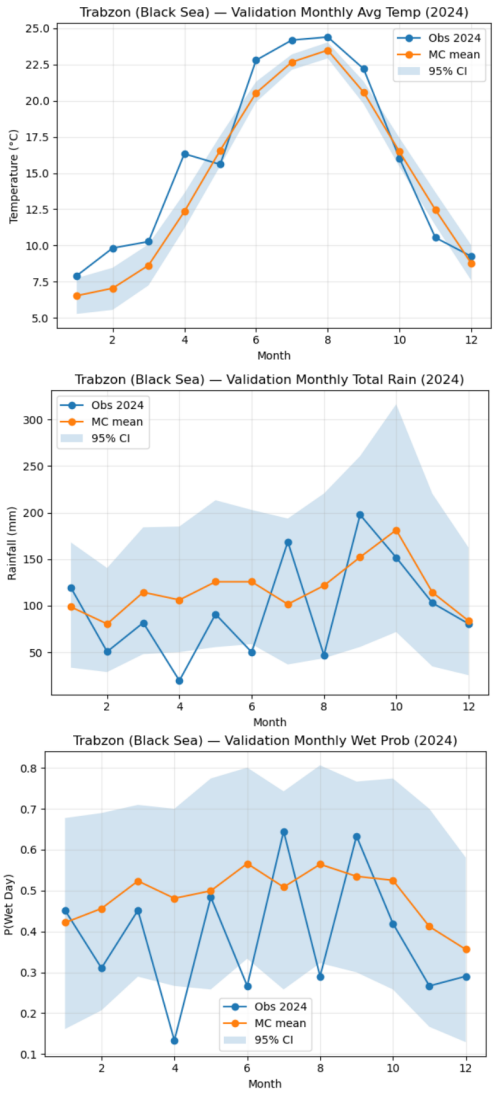



**Figure 21** Validation of monthly average temperature, total precipitation, and wet-day probability for Trabzon (Black Sea Region) in 2024

Figure 23 presents validation results for Diyarbakır, representing the semi-arid Southeastern Anatolia Region. Observed monthly mean temperatures closely align with the Monte Carlo ensemble mean and remain within the simulated uncertainty bounds, with particularly stable behavior during the hot summer months. Precipitation-related indicators exhibit a strong seasonal contrast. Wet-day probability and precipitation totals approach near-zero levels during summer, confirming persistent regional aridity, while precipitation occurrence and intensity increase during winter months. Wider uncertainty bounds in winter reflect increased variability linked to episodic rainfall events, whereas narrow summer bounds indicate statistically stable dry conditions. Overall, these results confirm that the stochastic framework successfully captures both the dominant summer dryness and the seasonal concentration of rainfall typical of Southeastern Anatolia.

To complement the visual comparisons between Figures 17 to Figure 23 and Table 9 provides a quantitative summary of model performance across all seven representative regions of Türkiye. The table reports simulated annual mean temperature, total precipitation, and number of wet days for 2025, together with out-of-sample validation errors computed using observed 2024 monthly indicators. Model accuracy is assessed using Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) for monthly average temperature, monthly total precipitation, and wet-day probability. Overall, temperature errors remain consistently low across regions, confirming that the seasonal thermal cycle is reproduced reliably. Precipitation-related errors are comparatively larger, reflecting the inherently intermittent and stochastic nature of rainfall; however, the results remain within acceptable ranges for stochastic weather generator applications. These quantitative findings further support the robustness and generalizability of the proposed stochastic framework across diverse climatic regimes in Türkiye.

**Table 9** Annual summary of stochastic simulation results and validation metrics for the seven geographical regions of Türkiye

| Region | City | $T_{25}$ | $P_{25}$ | $D_{25}$ | $MAE_T$ | $RMSE_T$ | $MAE_P$ | $RMSE_P$ | $MAE_{Wp}$ | $RMSE_{Wp}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| Aegean | İzmir | 18.27 | 704.37 | 78 | 1.65 | 2.05 | 34.70 | 48.72 | 0.09 | 0.11 |
| Black Sea | Trabzon | 14.80 | 1403.11 | 176 | 1.66 | 1.92 | 42.64 | 50.00 | 0.14 | 0.18 |
| Central Anat. | Ankara | 12.98 | 420.96 | 81 | 1.43 | 1.94 | 17.33 | 21.39 | 0.11 | 0.14 |
| Eastern Anat. | Erzurum | 6.90 | 612.92 | 118 | 1.57 | 2.02 | 36.41 | 46.41 | 0.09 | 0.12 |
| Marmara | Kocaeli | 14.96 | 865.14 | 129 | 1.54 | 1.84 | 34.91 | 40.40 | 0.13 | 0.16 |
| Mediterranean | Antalya | 18.78 | 1056.93 | 81 | 1.64 | 2.10 | 45.65 | 54.06 | 0.13 | 0.15 |
| S. Eastern Anat. | Diyarbakır | 16.66 | 586.58 | 79 | 1.26 | 1.53 | 16.93 | 25.42 | 0.08 | 0.11 |

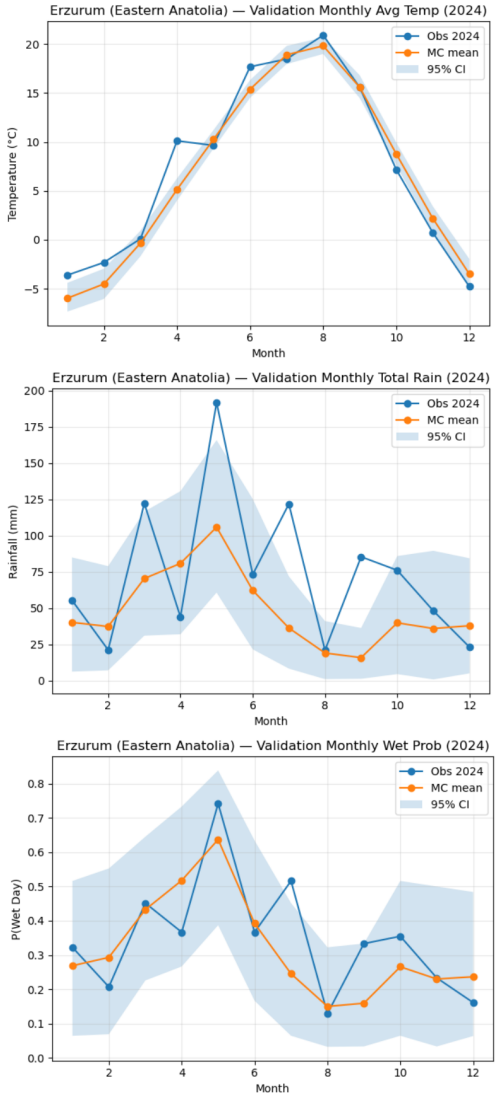



**Figure 22** Validation of monthly average temperature, total precipitation, and wet-day probability for Erzurum (Eastern Anatolia Region) in 2024

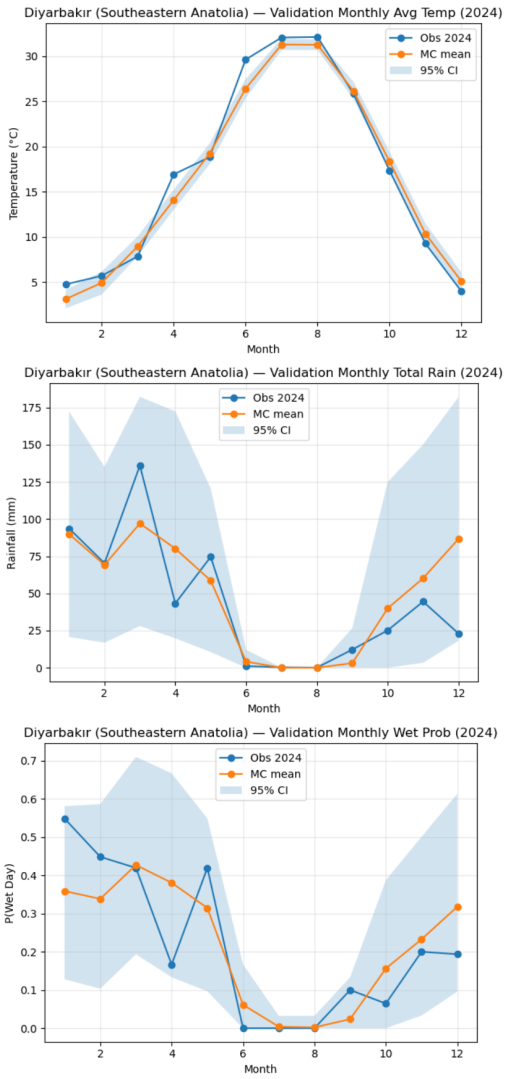



**Figure 23** Validation of monthly average temperature, total precipitation, and wet-day probability for Diyarbakır (Southeastern Anatolia Region) in 2024

## CONCLUSION

In this study, a Monte Carlo-based stochastic simulation framework was developed and implemented for representative cities across the seven geographical regions of Türkiye. The model successfully synthesized weather data, yielding regionally consistent temperature distributions and precipitation patterns. By

leveraging the Monte Carlo approach, atmospheric uncertainty was quantified through probability distributions, allowing for a robust assessment of potential climatic variations.The seasonal behaviors of the 95% confidence intervals align closely with the unique climatic characteristics of each region, confirming that the model provides a reliable representation of regional weather dynamics. Given the inherently stochastic nature of atmospheric processes, this approach captures the natural variability that deterministic models might overlook. Ultimately, the proposed integrated framework offers a computationally efficient and reliable tool for regional climate assessment. The out-of-sample validation using 2024 observations, followed by scenario generation for 2025, further supports the robustness of the model. These findings demonstrate the model's high potential for integration into climate-sensitive decision-support systems, such as agricultural planning, water resource management, and energy infrastructure modeling. Future studies could expand this framework by incorporating non-parametric distributions or multi-site spatial correlations to further refine regional climate risk assessments.

### Ethical standard

The authors have no relevant financial or non-financial interests to disclose.

### Availability of data and material

The data that support the findings of this study are available from the corresponding author upon reasonable request.

### Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## LITERATURE CITED


Ailliot, P., D. Allard, V. Monbet, and P. Naveau, 2015 Stochastic weather generators: an overview of weather type models. Journal de la société française de statistique **156**: 101–113.

Guan, X., V. D. Nguyen, P. Voit, B. Merz, M. Heistermann, *et al.*, 2025 The ability of a stochastic regional weather generator to reproduce heavy-precipitation events across scales. Natural Hazards and Earth System Sciences **25**: 3075–3086.

Kilsby, C. G., P. Jones, A. Burton, A. Ford, H. J. Fowler, *et al.*, 2007 A daily weather generator for use in climate change studies. Environmental Modelling & Software **22**: 1705–1719.

Knutti, R., M. R. Allen, P. Friedlingstein, J. M. Gregory, G. C. Hegerl, *et al.*, 2008 A review of uncertainties in global temperature projections over the twenty-first century. Journal of Climate **21**: 2651–2663.

Najibi, N., A. J. Perez, W. Arnold, A. Schwarz, R. Maendly, *et al.*, 2024 A statewide, weather-regime based stochastic weather generator for process-based bottom-up climate risk assessments in california–part i: Model evaluation. Climate Services **34**: 100489.

Nandan, R., V. Bandaru, P. Meduri, C. Jones, and R. Lollato, 2024 Evaluating the utility of weather generators in crop simulation models for in-season yield forecasting. Agricultural Systems **220**: 104082.

Palmer, T., G. Shutts, R. Hagedorn, F. Doblas-Reyes, T. Jung, *et al.*, 2005 Representing model uncertainty in weather and climate prediction. Annu. Rev. Earth Planet. Sci. **33**: 163–193.

Palmer, T. N., 2000 Predicting uncertainty in forecasts of weather and climate. Reports on progress in Physics **63**: 71.

Schuol, J. and K. Abbaspour, 2007 Using monthly weather statistics to generate daily data in a swat model application to west africa. Ecological modelling **201**: 301–311.

Semenov, M. A., 2008 Simulation of extreme weather events by a stochastic weather generator. Climate Research **35**: 203–212.

Semenov, M. A., R. J. Brooks, E. M. Barrow, and C. W. Richardson, 1998 Comparison of the wgen and lars-wg stochastic weather generators for diverse climates. Climate research **10**: 95–107.

Wilks, D., 1999a Simultaneous stochastic simulation of daily precipitation, temperature and solar radiation at multiple sites in complex terrain. Agricultural and Forest Meteorology **96**: 85–101.

Wilks, D. S., 1998 Multisite generalization of a daily stochastic precipitation generation model. journal of Hydrology **210**: 178–191.

Wilks, D. S., 1999b Interannual variability and extreme-value characteristics of several stochastic daily precipitation models. Agricultural and forest meteorology **93**: 153–169.

Woodson, D., S. Gangopadhyay, L. Bearup, A. Verdin, E. Shamir, *et al.*, 2025 wxgenr: An r package for stochastic weather generation with seasonality. SoftwareX **31**: 102209.

Zippenfenig, P., 2023 Open-meteo.com weather api.



***How to cite this article:*** Kılıç, N. H., and Sevin, A. Stochastic Weather Simulation of Türkiye's Geographical Regions: A Monte Carlo Framework. *ADBA Computer Science*, 3(1), 37-50, 2026.




ADBA Computer Science
Cutting-Edge Scientific Solutions

e-ISSN: 3023-8544
RESEARCH ARTICLE
Vol.3 / No.1 / 2026 / pp.51-56
https://doi.org/10.69882/adba.cs.2026017

# Design of an ESP32-Based Smart Meteorological Data Collection Station for Renewable Energy Applications

**Bedir Kaan Karapapak** [*,1], **Mevlüt Eren Küçüker** [*,2], **Mert Ali Topal** [*,3], **Eren Emek** [*,4] **and Mohammad Soukar** [*,5]
*Hitit University, Faculty of Engineering and Natural Sciences, Department of Electrical and Electronics Engineering, 19030, Çorum, Türkiye.

**ABSTRACT** This study presents the design and implementation of an ESP32 microcontroller based, modular smart meteorological data collection station developed to increase the efficiency of renewable energy systems and monitor environmental data at a low cost.The system is designed as an alternative to high-cost industrial stations. Core hardware components include BMP280 (temperature, pressure), wind speed (Hall Effect anemometer), and wind direction (Hall Effect wind vane) sensors. ESP32 is used as the central control unit with its built-in Wi-Fi/Bluetooth features; it processes data in string format and transmits it to a Python-based desktop interface via Bluetooth serial communication protocol (at a frequency of 1 Hz). Additionally, a one-week circular storage logic (Circular Buffer) was created using the LittleFS file system on the ESP32 for uninterrupted data storage. A solar panel and battery management system were designed for off-grid operation capacity. As a result of validation tests, it was proven that the system provides high stability in data transmission and that linear regression-based calibration (over 50% error improvement) is mandatory to reach professional standards, especially in atmospheric pressure measurements. This low-cost and energy-efficient platform aims to provide a scalable, domestic data collection infrastructure for renewable energy sites and smart agriculture projects.

**KEYWORDS**
Weather station
Internet of things
LittleFS
ESP32
Anemometer
Wind vane
Renewable energy
Smart sensors

## INTRODUCTION

Monitoring changes in meteorological data is of great importance both for agricultural activities and for the efficient use of renewable energy sources such as solar and wind. Reliable meteorological data are needed for accurate site analysis, especially during the installation phase of wind and solar power plants. However, the high costs of industrial-type weather stations limit the widespread use, development, and use of these systems by individual researchers.

Internet of Things (IoT) technology, which has gained a place among today's technologies and continues research and development activities in many application areas, basically enables real-world objects to communicate over the internet through different technologies. In other words, IoT is used to connect the real world to the internet. IoT systems generally include hardware such as sensors, actuators, and smart devices. In IoT-based applications, functions for remote control, monitoring, and operation of systems are implemented. Within the scope of this study, a low-cost, ESP32 microcontroller-based smart meteorological data collection station has been designed and implemented.

Instant tracking and analysis of meteorological data are critical for increasing agricultural productivity, disaster management, and especially for the efficient use of renewable energy sources in line with the increasing energy demand in recent years. The high installation costs and maintenance difficulties of traditional weather stations have led researchers to develop microcontroller-based, modular, and low-cost solutions.

When the literature is examined, it is seen that studies in this field generally vary around the processor architecture used, energy management strategies, and data transmission protocols. In studies on the adaptation of industrial standards in data collection and transmission, the integration of microcontroller-based systems with wireless sensor networks draws attention. For example, in a portable station designed using a PIC microcontroller, the Modbus protocol was preferred for collecting and transmitting sensor data. In this system, data were transferred to the main center via Zigbee wireless modules and serial interfaces, creating a hybrid communication structure (Devaraju *et al.* 2015).

In applications where processing power and data storage capacity are at the forefront, the use of single-board computers is common. In studies where Raspberry Pi-based systems were developed, wind speed, direction, temperature, and pressure data were processed, stored in local databases, and visualized via web-based

**Manuscript received:** 13 November 2025,
**Revised:** 27 December 2025,
**Accepted:** 9 January 2026.

[1]bedirkaank@gmail.com (**Corresponding author**)
[2]mevluterenkucuker@gmail.com
[3]meertalitopal@gmail.com
[4]erenemek2004@gmail.com
[5]soukarmuhammed@gmail.com

interfaces. While these systems offer high data processing capacity, they may pose a disadvantage in terms of energy consumption in off-grid sites (Baste and Dighe 2017; Üçgün *et al.* 2021).

Energy independence is one of the most important factors for the sustainability of the system in field studies. Research focusing on this problem shows the use of solar-powered microcontrollers with optimized power consumption. In such systems, by establishing SMS-based warning mechanisms via GSM modules, data flow could be ensured even in regions without internet infrastructure. In validation tests, it has been reported that such low-cost systems provide a high degree of compliance with national meteorological data (Haque *et al.* 2019).

In current studies, the ESP32 microcontroller and IoT platforms, which stand out with both low power consumption and built-in Wi-Fi/Bluetooth features, have started to be preferred. Modern approaches have been exhibited in systems developed using ESP32, where data is instantly transferred to cloud-based platforms and can be tracked via mobile devices (Murthy *et al.* 2023). Furthermore, it has been proven that barometric pressure sensors (such as BMP280) used in such low-cost systems can produce data with accuracy very close to reference devices when calibrated with statistical analysis and linear regression methods (Kusuma *et al.* 2023).

On the other hand, comprehensive systems that measure not only standard meteorological data but also specific parameters such as sky quality and astronomical vision are available in the literature. Autonomous station prototypes such as "GÖKSİS," developed in a nationally conducted project, can collect very sensitive data using advanced optical and thermal sensors. However, the cost and installation difficulty of such systems restrict accessibility for individual users or small-scale power plants (Yerli *et al.* 2017).

In light of the existing literature, there is a need for a domestic and low-cost system that both works compatibly with renewable energy sources (solar/wind) and incorporates current IoT technologies. The platform proposed in this study aims to fill this gap in the literature with its modular structure built on the ESP32 architecture and its energy efficiency-oriented design, offering a cost-effective solution especially for renewable energy sites.

## MATERIALS AND METHODS

In this study, a low-cost, portable, and modular data collection station was designed for use in areas where renewable energy systems are utilized and at meteorological observation points. In the development of the system, an integrated method consisting of hardware component selection, circuit design, embedded software development, and data analysis stages was followed.

### Hardware Design and Components

The hardware components used in the system were selected to be suitable for operating in harsh environmental conditions and low power consumption criteria.

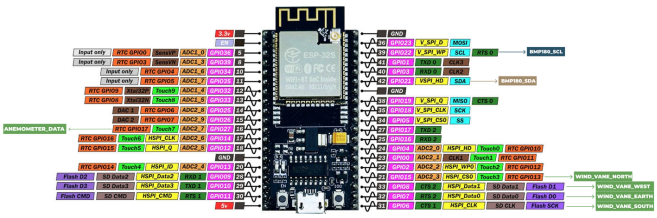


**Figure 1** Esp32 Connection Diagram

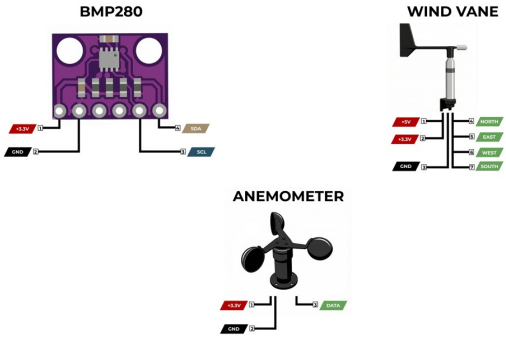



**Figure 2** Esp32 Connection Diagram

***Central Control Units(ESP32S):*** Although Arduino or PIC-based microcontrollers are often preferred in low-cost weather station designs designed by researchers, the ESP32 development board was preferred in this study due to Wi-Fi/Bluetooth-based data transmission requirements and the use of a modern user interface. Thanks to its dual-core processor architecture and built-in Wi-Fi/Bluetooth modules, data can be easily stored directly on the user's device or in the cloud without the need for additional communication hardware. Additionally, data can be stored in 1-week periods thanks to the ESP32's LittleFS file system.

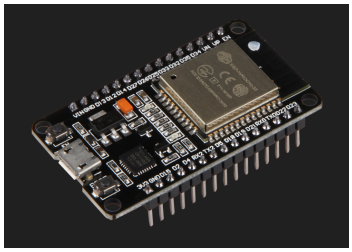


**Figure 3** ESP32S (joy-iT.net 2024)

***Meteorological Sensors:*** The sensors used in the station were selected according to measurement principles proven reliable in the literature:

- **Atmospheric Pressure Sensor (BMP280):** The BMP280 sensor, which uses piezoresistive technology to measure temperature, air pressure, and altitude, has been integrated into the system. Statistical analyses in the literature show that when this sensor is calibrated with linear regression models, it provides data with accuracy very close to reference devices and is a reliable measurement tool.

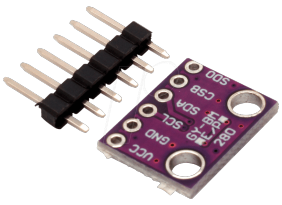


**Figure 4** BMP280 (reichelt.com 2024)

- **Wind Speed Sensor(Anemometer):** A 3-cup anemometer, which produces pulses depending on the rotation speed, was

used to measure wind speed. To measure wind speed accurately, a Hall Effect-based measurement system that converts mechanical rotation into electrical signals was designed. With each rotation of a permanent magnet placed on the anemometer's shaft, the Hall sensor detects the magnetic field and produces a digital pulse. This method minimizes friction, allowing stable measurements even at low wind speeds. The Hall sensor used in the system is powered by 5V for stable operation and signal integrity. However, the input-output (I/O) pins of the microcontroller unit (ESP32) where data is processed operate at a 3.3V logic level. Applying the 5V output signal directly to the microcontroller carries the risk of hardware damage and data errors. To eliminate this incompatibility and ensure system safety, a Logic Level Shifter was integrated between the sensor output and the microcontroller input. This circuit performs the signal conditioning process by reducing 5V amplitude square wave signals to 3.3V without data loss. The obtained pulse signals are converted to wind speed (m/s) based on the number of pulses per unit time.

**Figure 5** Three-Cup Anemometer Design and Electronic Enclosure Box.

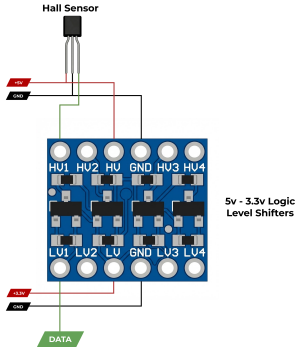



**Figure 6** Hall effect sensor and Logic Level Shifter connection diagram.

- **Wind Direction Sensor (Wind Vane):** To obtain the wind direction data of the weather station, a mechanical wind vane and a magnetic-based sensing system that converts the movement of this vane into digital data were designed. For direction determination, four Hall Effect sensors placed at 90-degree angles to represent the main directions (North, East, West, South) were used in the system. The working principle of the system is based on a permanent magnet connected to the shaft of the wind vane triggering the sensor in the relevant direction during rotation. As seen in Figure 6, the sensors are placed in an "N, E, W, S" (North, East, West, South) sequence, dividing the 360-degree scanning area into four main regions. Whichever sensor the magnet enters the field of, that sensor produces a "low" or "high" signal, informing the microcontroller of the direction the wind is blowing from.

**Figure 7** Wind Vane and Electronic Enclosure Box.

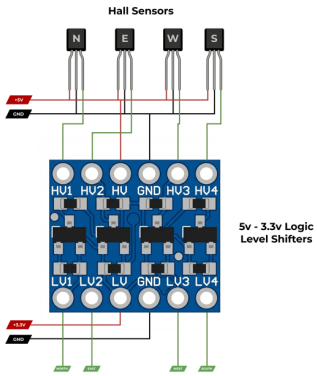



**Figure 8** Hall effect sensors and Logic Level Shifter connection diagram.

***Power Management System:*** A renewable energy-based power supply unit has been designed so that the system can operate uninterruptedly in off-grid areas. This unit consists of a solar panel, a charge control circuit, and a battery group. As emphasized in similar studies, this structure allows the system to collect data autonomously in rural areas and ensures energy sustainability.

### Software Architecture and Data Transmission

The software infrastructure of the system consists of two main components: the embedded software on the ESP32 side and the Python-based interface software on the computer side. The data flow is carried out with a certain hierarchy from the physical layer to the user screen.

***Data Transmission Protocol and Packet Structure:*** Data transmission in the system is designed in a text-based (string) structure to reduce complexity and facilitate debugging processes. Sensor data read by the ESP32 are combined into a single character string using a specific separator (e.g., comma or semicolon).

- **Data Format:** Each data packet is structured in the format: `Temp;Humidity;Pressure;Wind_S;Wind_D`
- **Transmission Channel:** These prepared character strings are transmitted to the Python-based interface software once per second (1 Hz) via the Bluetooth serial communication protocol.

***Python-Based Desktop Application:*** The user interface was developed using the Python language, which is rich in terms of data science and library support. The software consists of the following functional blocks:

- **Data Listening and Parsing:** String data coming through Python's pyserial library is captured. The texts separated by the split() function are converted into numerical data (float/int) and transferred to the graphics engine.
- **Visualization and Analysis:** Incoming instantaneous data are shown in real-time on temperature, humidity, and pressure trend graphs (Temperature Trend, Humidity Trend, etc.). Additionally, the system calculates instantaneous minimum, maximum, and average values and presents them to the user.
- **Historical Data Integration:** The user interface can communicate asynchronously with the LittleFS unit on the ESP32. When the user gives the "Get Historical Data" command, the ESP32 transfers the recorded string data of the last week to the Python interface, and these data are displayed in historical analysis graphs.


**Figure 9** Python-Based User Interface Designed Over Bluetooth Communication.

***Error Control and Stability:*** A data validation mechanism has been established on the Python software to prevent data loss or faulty packet transmission that may occur during communication. By checking the length and format of the incoming string packet, non-standard data is prevented from being reflected in the graphs, thus protecting the stability of the system's operation.

### Validation and Test Method

To measure the reliability and sensitivity of the system, a three-stage testing process was applied: sensor data accuracy, data transmission stability, and storage unit consistency.

***Sensor Accuracy and Calibration Test:*** The data produced by the sensors used in the system (BME280 and Anemometer) were verified by comparing them with reference values.

- **Thermodynamic Data Validation:** Temperature, humidity, and pressure values were verified according to pre-calibrated standard sensor data and re-calibrated.
- **Wind Data Test:** Wind speed and direction data received from the anemometer were tested under controlled airflow, and the response time of the wind vane and trend graphs in the Python interface was measured.
- **Error Margin Analysis:** Deviation rates of the values received from the sensors from the reference values were calculated, and it was confirmed that the system operates within acceptable error limits.

***Data Transmission and Communication Tests:*** The stability of string-based data transmission via Bluetooth was tested.

- **Packet Loss Analysis:** At 115200 baud rate, the arrival rates of data packets sent at certain distances (1m, 5m, 10m) to the Python interface were checked.
- **String Parsing Test:** It was verified with outlier tests that the string expressions sent from the ESP32 were correctly split and converted into numerical data on the Python side.
- **Connection Continuity:** The continuous operation time (uptime) of the system after the "Connect and Start" command was observed, and the stability of the Bluetooth connection was measured.

***Storage and Data Integrity Tests:*** The data storage capabilities of the LittleFS file system and the one-week restriction logic were tested.

- **Write/Read Test:** Power interruption simulations were performed to check whether sudden shutdowns caused damage to historical data on LittleFS.
- **Weekly Cycle (Circular Buffer) Validation:** Flash memory occupancy was monitored with accelerated sampling tests; it was verified that when the one-week period was full, the oldest data was deleted and new data was written over it (FIFO - First In First Out).
- **Recall Performance:** The transfer speed of historical data requested via the Python interface over Bluetooth and the process of reflecting it correctly on the graphs were tested.

## RESULTS

In this study, an ESP32 microcontroller-based, low-cost, and high-efficiency modular weather station design and implementation were successfully carried out. Both the hardware and software layers of the system were optimized for reliable collection, local storage, and visualization of meteorological data. The main findings obtained as a result of the study are as follows:

- **Data Management and Storage:** Thanks to the LittleFS file system configured on the ESP32, meteorological data were safely stored in local memory in one-week periods. It was observed that no data loss occurred in cases such as power cuts, and the circular storage logic successfully prevented memory overflows.
- **Communication Performance:** Converting the data received from the sensors into string format and transferring it at 115200 baud rate via Bluetooth protocol provided high stability in data transmission. The performance of the Python-based user interface in parsing incoming packets in real-time and converting them into visual graphs increased the traceability of the system.
- **Sensor Accuracy and Calibration:** Comparisons with similar studies in the literature, Kusuma *et al.* (2023) proved that BMP280/BME280 series sensors are quite sensitive in barometric pressure measurements, but linear regression-based calibration is mandatory to reach professional measurement standards (BMKG, etc.). In the analyses performed, it was determined that the error margin (RMSE) could be improved by more than 50% after calibration.

- **User Interface and Analysis:** Thanks to the developed desktop software, instantaneous weather tracking, wind direction, and speed analysis, and historical data query operations were presented to the user through a single platform.

**Figure 10** General view of the developed modular meteorological station hardware, including the anemometer, wind vane, and the main control unit.

### Measurement Accuracy and Calibration Performance

The most basic success criterion of the system is the accuracy of the collected data118. In studies where BMP280 sensors are used in the literature, it has been observed that error rates (RMSE) drop significantly when data are compared with reference devices and calibrated with linear regression analysis. The station developed within the scope of this project is expected to show a similar performance. After the calibration process, it is envisaged that a high correlation ($R^2 > 0.98$) will be achieved with the reference station in atmospheric pressure measurements, and measurement deviations will remain within acceptable limits ($\pm 1$ hPa for pressure, $\pm 0.5$ °C for temperature). In wind speed and direction measurements, it is aimed to convert instantaneous changes into digital data with over 95% accuracy thanks to analog signal processing algorithms.

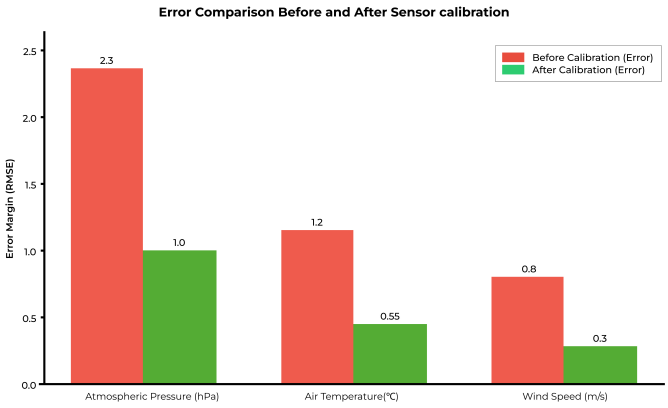



**Figure 11** Comparison of RMSE values before and after sensor calibration for atmospheric pressure, air temperature, and wind speed.

## CONCLUSION

The verification of the off-grid operation capacity of the system is another important expected result. Thanks to the solar panel and battery management system, the station is aimed to be able to collect data continuously for at least 48 hours even on cloudy days. By using the deep-sleep modes of the ESP32 effectively in software, it is expected that the average power consumption of the system will be minimized, thus ensuring energy continuity. Additionally, the outer case to be designed to protect the hardware against environmental factors (rain, dust, humidity) is expected to ensure the longevity of the system.

The developed prototype is expected to be produced at a much lower cost (approximately 1/5 ratio) compared to industrial weather stations with similar features. In conclusion, this developed system is aimed to contribute to the literature in the field of domestic data collection systems and to create a scalable infrastructure for future "smart agriculture" or "smart solar farm" projects. In future studies, it is planned to further develop the modular structure by powering the system with solar energy and integrating long-range communication protocols such as LoRaWAN.

### Ethical standard

The authors have no relevant financial or non-financial interests to disclose.

### Availability of data and material

Not applicable.

### Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## LITERATURE CITED


Baste, P. and D. Dighe, 2017 Low cost weather monitoring station using raspberry pi. International Research Journal of Engineering and Technology (IRJET) **4**: 3184–3189.

Devaraju, J. T., K. R. Suhas, H. K. Mohana, and V. A. Patil, 2015 Wireless portable microcontroller based weather monitoring station. Measurement **76**: 189–200.

Haque, M. I., A. H. M. D. Shatil, A. N. Tusar, M. Hossain, and M. H. Rahman, 2019 Renewable powered portable weather update station. In *2019 International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST)*, pp. 374–377.

joy-iT.net, 2024 NodeMCU ESP32. https://joy-it.net/en/products/SBC-NodeMCU-ESP32, Accessed: October 21, 2025.

Kusuma, H., D. Oktavia, S. Nugraha, T. Suhendra, and S. Refly, 2023 Sensor bmp280 statistical analysis for barometric pressure acquisition. IOP Conference Series: Earth and Environmental Science **1148**: 012008.

Murthy, M. S., R. P. R. Kumar, B. Saikiran, I. Nagaraj, and T. Annavarapu, 2023 Real time weather monitoring system using iot. In *E3S Web of Conferences*, volume 391, p. 01142, EDP Sciences.

reichelt.com, 2024 FREI Developer Boards - Temperature and Pressure Sensor, BMP280. https://www.reichelt.com/de/en/shop/product/developer_boards_-_temperature_and_pressure_sensor_bmp280-266034, Accessed: October 21, 2025.

Yerli, S. K., T. Ak, N. Aksaker, V. Bakış, S. Evren, *et al.*, 2017 Meteoroloji destekli astronomik görüş Ölçümü İçin otomatik İstasyon tasarımı ve prototip yapımı. Tübitak Projesi, Orta Doğu Teknik Üniversitesi, Fen Edebiyat Fakültesi, Ankara .

Üçgün, H., Z. K. Kaplan, and U. Yüzgeç, 2021 Akıllı hava İstasyonu ile iot tabanlı hava durumu İzleme sistemi. Avrupa Bilim ve Teknoloji Dergisi pp. 563–571.

***How to cite this article:*** Karapapak, B. K., Küçüker, M. E., Topal, M. A, Emek, E., and Soukar, M. Design of an ESP32-Based Smart Meteorological Data Collection Station for Renewable Energy Applications. *ADBA Computer Science*, 3(1), 51-56, 2026.



ADBA Computer Science
Cutting-Edge Scientific Solutions

e-ISSN: 3023-8544
RESEARCH ARTICLE
Vol.3 / No.1 / 2026 / pp.57-62
https://doi.org/10.69882/adba.cs.2026018

# A Systematic Benchmark of Advanced Architectures for Automated Multi-Class Classification in Digital Pathology

**Mehmet Ali Eryuksel** *,1, **Yigitcan Cakmak** *,2 **and Ishak Pacal** *,β,3

*Department of Computer Engineering, Faculty of Engineering, Igdir Universty, 76000, Igdir, Türkiye, βDepartment of Electronics and Information Technologies, Faculty of Architecture and Engineering, Nakhchivan State University, AZ 7012, Nakhchivan, Azerbaijan.

**ABSTRACT** Histopathological assessment of tissue biopsies is the main way to diagnose breast cancer. The current truth is that interpreting histopathological images is subjective and typically requires a great deal of effort by busy pathologists. Deep learning has transformed the practice of Digital Pathology, but at this moment, there is no universal agreement on which architecture gives the best performance for multi-class tissue recognition. The goal of this work is to analyze and compare the traditional Convolutional Neural Network (CNN), ResNet-101 and DenseNet-169, to the recently developed Transformer architecture, the Vision Transformer (ViT), by using a systematic benchmarking approach. Our approach involved using a balanced dataset with images from four classes (Benign, InSitu, Invasive, and Normal) and included preparation of images to a standardized input size of 224x224, transfer learning, and standard augmentations. Experimental results indicated that DenseNet-169 performs significantly greater than ResNet-101 (75% accuracy) with an improved accuracy of 96.25% and F1-score of 0.9628 at comparatively low levels of computational power (67.169 GFLOPs). DeiT Base is also an effective diagnostic adjunct, but due to its extensive number of parameters (85.80M) and computational cost, there are clear advantages in using optimized dense CNN architectures in limited clinical resources.

**KEYWORDS**
Convolutional neural networks
Vision transformers
Multi-class classification
Comparative study
Computer-aided diagnosis
Deep learning
Histopathology
Breast cancer

## INTRODUCTION

Breast cancers rank among the major healthcare issues across the globe, and they contribute to a large percentage of cancer-related deaths among women (Ma *et al.* 2024; Akbari *et al.* 2026; Alkhafaji *et al.* 2026). The present scenario regarding breast cancers, as depicted above, has continued to pose significant healthcare threats to women across the world, and their comprehensive and accurate identification still remains a challenge. Presently, for accurate identification of cancers, the "Gold Standard Test," or properly referred to, still remains the Histopathological examination of tissue biopsies. The pathologists, after careful observation of these slides, are able to differentiate the tissue patterns, viz., Benign, InSitu, Invasive, and Normal, and so on. But this needs to be done manually and hence is time-taking, to say the least, and still tends to be a challenge due to possible variations (Yan *et al.* 2025; Jahan *et al.* 2025).

In recent years, with the introduction of deep learning into digital pathology, there was a major change in thinking and how pathway classification would be handled. Convolutional Neural Networks (CNNs) have been the dominant force in this area, due mainly to their ability to extract spatial hierarchies and local features from high-resolution digital medical imaging (Yu *et al.* 2024; Rahman *et al.* 2025; Irmak and Saygılı 2025). A variety of CNN architectures, including ResNet-101 and DenseNet-169, have provided many examples of strong benchmark architectures, addressing the problem of gradient degradation, and promoting the concept of feature reuse. With the continued development of digital pathology comes the introduction of Vision Transformers (ViTs) as well as hybrid architectures, such as the newly developed Data-efficient Image Transformer (DeiT). ViTs and DeiT also allows for

**Manuscript received:** 14 October 2025,
**Revised:** 8 January 2026,
**Accepted:** 9 January 2026.

[1]m.alieryuksel78@gmail.com
[2]ygtcncakmak@gmail.com(**Corresponding author**)
[3]ishakpacal@igdir.edu.tr

the integration of global context into classification models and, in particular, enable the recognition of small levels of pathologic change across an entire slide (Hasan *et al.* 2025; Sumitha and Isaac 2025).

Despite such advances, an important issue is still remaining, which is a balance between high detection performance and efficient computation (Maurya *et al.* 2024; Singh and Patnaik 2026; Ahmed 2025). Although more advanced yet complex models could achieve even superior performance, their applicability in a practical environment, such as a hospital where hardware or computational capacity could be more restricted, would still include a careful consideration of parameters and GFLOPs (Kansal *et al.* 2025). Moreover, previous research works are mostly concentrating on a binary classification problem, like malignancy versus benign-nous cases, although, in reality, a more appropriate classification is needed for multi-class problems that are commonly experienced by a human pathologist on a daily basis. There is an obvious gap between existing research works concerning a benchmark of multi-class BC using traditional CNN models and more contemporary models like transformers (Singh and Patnaik 2025; Zhang *et al.* 2025). This study is intended to address this shortcoming by presenting an overall assessment of multiple SOTA models, including DeiT Base, DenseNet-169, and ResNet-101, which have been tested with a carefully curated set of four histopathological classifications. They were evaluated based on precision, recall, f1 score, and computational burden.

## RELATED WORKS

Computational Intelligence has transformed the experience of digital pathology from one of subjectively evaluating using a visual microscope to one of objective, high Throughput Diagnostic Systems for all cancers. The greatest complication in classifying breast cancer automatically is the great differences in morphology within tissue samples as benign and malignant cell types typically contain similar cellular characteristics. CNNs have historically been utilized to extract spatial features from tissue samples today there are increasing developments toward hybrid systems, frequency domain analysis and explainable A.I. (XAI) which makes A.I. more interpretable and efficient while still enabling high levels of Predictive Performance.

The study by Liu *et al.* (2024) discussed the limitations of existing methods used for feature representation in digital pathology. Liu et al. proposed a novel method for adaptive threshold selection which can be applied to histopathological images processed in the frequency domain using discrete wavelet transform. This adaptive thresholding process removes unnecessary noise and enhances the relevant textural features of histopathological images that are critical for producing high-resolution histopathological images. The results from their study show that adding threshold selection as part of the model's training improves convergence on complex tissue structures.

According to Gu *et al.* (2024), deep learning has been systematically evaluated to assess the role of deep learning for automating breast cancer diagnosis using automated systems. The authors highlight the value of improving the quality of preprocessing and optimising hyperparameters to achieve the most accurate diagnoses possible from these systems. The authors denote that although new architectures have value, they also suggest that utilising existing networks with strong augmentation techniques provide opportunities for diagnostic accuracy results that are comparable or superior to those achieved by newer architectures or complex tailored architectures. This research supports the need for standardisation of digital pathology workflow processes in order to reproduce results in multiple clinical environments.

In a supervised contrastive learning framework (CHistNet) proposed by Rahaman *et al.* (2024), the authors aim to increase the ability for distinguishing features in multi-class histopathology. This is accomplished through the clustering of related pathologies in latent space while maximizing separation of unrelated classes. This method is highly effective at alleviating class imbalances that exist in breast oncology. Their research demonstrates how contrasting the results from using contrastive strategies produces a more optimal representation of the features than training using only traditional cross-entropy methods.

The researchers (Krishnappa and Reddy 2024) developed a new kind of deep ensemble graph network (DEGN) that treats the spatial organization of tissues as a series of related nodes instead of a two-dimensional pixel grid. This allows the DEGN to better represent long-range spatial relationships in a tissue sample and therefore produce a more complete understanding of the tissue microenvironment. The authors also demonstrated that an ensemble method outperforms standard CNNs at distinguishing between small architectural differences in breast tissues.

In a study conducted by Singh *et al.* (2024), colleagues explored how effective transfer learning is for classifying digital histopathological images. They specifically sought to address the limitations of relying on limited amounts of annotated digital histopathology images. Specifically, through fine-tuning a large number of pretrained deep learning models specifically designed for histopathology tasks, they demonstrated that they could accurately distinguish between multiple grades of breast tumour types (defined by the World Health Organisation) while decreasing the time it took to train these neural networks substantially. Singh's team's work demonstrated that transfer learning represents a model that is easily scalable for implementing AI-based diagnostics in low-resource healthcare settings.

Yilmaz *et al.* (2025) Concluded through an extensive review (n = 30) of advanced neural network architectures (CNNs and ViTs) focusing on the detection of breast cancer from ultrasound and histological images. They found that although multi-scale ViTs such as MViTv2-Base are able to exploit large context globally, they are most efficient when combined with lightweight CNNs (MobileNetV3). When choosing the correct architecture, this benchmark provides guidance based on the various imaging modalities and the appropriate diagnostic need.

To overcome the issue of interpreting results from Deep Learning methods, the authors (Ukwuoma *et al.* 2025) proposed a classification strategy based on Explainable AI (XAI) using LIME and SHAP technologies to demonstrate how their algorithm provides a reliable explanation of tissue types. The framework accurately classifies the tissue types and also produces a visual explanation of how the algorithm arrived at its decision by providing a clear visual representation of the nuclear or stromal features that influenced the predictive process of the algorithm. The development of the framework helps fill the gap between the predictions made by algorithms and the reasoning that is understandable by humans, creating a bridge to building confidence in AI-assisted diagnostic devices within clinical practice and obtaining the requisite regulatory approvals for the use of AI-based diagnostic tools within clinical practice.

## MATERIALS AND METHODS

### Dataset and Data Preprocessing

In our research, we employed a broad histopathology dataset, which comprises four types of tissue and covers the most difficult cases in diagnosing breast cancers, namely, Benign, InSitu, Invasive, and Normal. These categories, as depicted in Figure 1, are very varying in cell structures, density, and organization, and it is essential for the multi-class problem that these aspects are well picked up by the model. To ensure a balance in the training environment and avoid the overfitting issue by giving preference to a certain class, the dataset was selected to be well-represented by an equal number of 100 images for each class, resulting in a well-prepared set of 400 high-quality images. The level of complexity, as depicted in Figure 1, necessitates the needs for the advanced model to well-capture not only the local features but also the global features (Dataset 2025).

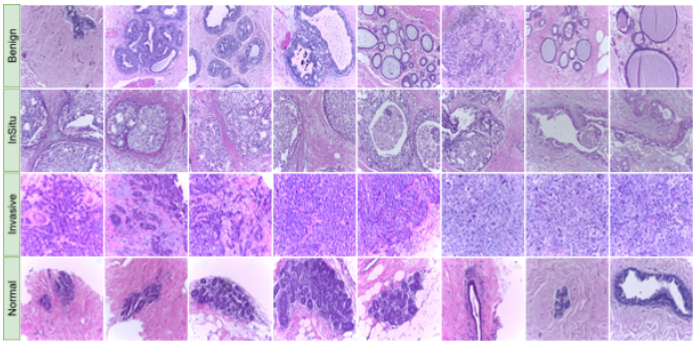



**Figure 1** Sample images from the breast histopathology classes in the dataset.

We implemented a consistent preprocessing procedure for all the data, before using it in the deep learning (DL) algorithms. A primary aspect of this process was that all the patches of the histopathology were resized to the same 224 x 224 pixels. These particular dimensions were selected to align with the input requirements of the existing architectures (e.g., DenseNet-169, DeiT Base). Thus they are suited for direct comparisons. The partitioning of each set into training, validation, and testing subsets (70/10/20 as illustrated in Table 1) allows models to train on sufficient data while maintaining a dedicated test set to accurately assess generalization capabilities on novel pathological patterns.

### Foundational Principles of Deep Learning Models (CNNs and ViTs)

Two different architectural philosophies have significantly impacted the development of automated histopathology: CNNs and VITs. While CNNs primarily focus on local morphology with their spatial hierarchies, VITs allow for a type of global understanding which has become increasingly important for identifying complex tissue architectures within the context of breast cancer diagnosis.

For more than 10 years, CNNs have been the standard for medical image analysis because of their unique ability to represent the local spatial correlation of histopathological slides. In this evaluation, we focus on two major advances in the architecture of CNNs to help mitigate the weaknesses of both ResNet-101 (He *et al.* 2025) and DenseNet-169 (Huang *et al.* 2025). ResNet-101 introduced the "skip connection" or residual learning concept that allows CNN's to avoid the vanishing gradient problem and learn the identity mapping across 101 layers, therefore maintaining important elements in the image. Alternatively, DenseNet-169 proposes that every layer in a block is connected; this allows the features from previous layers to be reused as much as possible while also reducing the number of parameters, thus still maintaining the interpretation power of Deep Learning. Therefore, CNNs, as a technique in Digital Pathology, can easily detect subtle changes in cells or nuclear pleomorphism in Benign or Invasive tissue patches.

Though CNNs use localized kernels, with the advent of ViTs, there is a paradigm shift towards understanding images in a more holistic manner. Unlike previous models, where an image is treated as a patch, in ViTs, an image is treated as a set of patches, and to understand their relevance, there is the use of the "self-attention" technique, where the relevance of each patch is measured in relation to another. Though conventional ViTs are very efficient, their main bottleneck in medical imaging is their requirement for huge quantities of data. To counter this, this work uses DeiT Base (Touvron *et al.* 2025). This model reduces the requirement for huge quantities of data with the help of its unique teacher-student distillation technique using distillation tokening, ensuring small quantities of data (as explained in Table 1) are efficiently utilized without sacrificing its global receptive width, making it the best-suited model for multi-classification problems where the positioning of the stroma and epithelium regions plays an important role in diagnosing whether the cell is of type InSitu or Normal.

### Transfer Learning and Data Augmentation

The lack of sufficient annotated examples in histopathology continues to be an obstacle in developing effective medical imaging datasets. To overcome this and achieve strong convergence in deep learning models, we used a Transfer Learning approach where instead of using resources to build our own networks from scratch (a process that would have been extremely costly compared to developing a dataset this size; see Table 1.), we took advantage of networks that were already trained on ImageNet-1K. This allowed us to use a large number of diverse features that were learned in advance by the networks to enable the models to concentrate on adjusting only the higher level portions of their architecture to identify the more specific characteristics of breast tissue, like the way that the structure of invasive carcinoma is different than the structure of breast tissue from in situ lesions.

For our Data Augmentation pipeline, we leveraged timm, a popular PyTorch Image Models library, where we utilized their standard stochastic transformations and avoided building an entirely new augmentation space from scratch. We took advantage of timm's "default" settings, which have demonstrated successful performance across modern deep learning benchmark tests. For example, the default settings incorporate key geometric and color-space perturbations, including random resized cropping, horizontal flipping, and basic color jittering, that affect a model's ability to generalize well to changes in scale or orientation (invariance). By using these standard default parameters, we ensured that any training environment we created for comparing the models (i.e., CNN vs. ViT) was repeatable and free from bias, allowing our analysis to focus primarily on the characteristics of each architecture (Wang *et al.* 2024; Mumuni *et al.* 2024).

### Metrics for Performance Assessment

To evaluate the effectiveness of the proposed DL models in the multi-class classification of breast cancer histopathology images, several standard quantitative metrics were employed to provide a rigorous assessment of their diagnostic performance. Accuracy was first utilized to offer a general measure of the models' abil-

**Table 1** Dataset Distribution by Class

| Class | Train (70%) | Val (10%) | Test (20%) | Total |
|---|---|---|---|---|
| Benign | 70 | 10 | 20 | 100 |
| InSitu | 70 | 10 | 20 | 100 |
| Invasive | 70 | 10 | 20 | 100 |
| Normal | 70 | 10 | 20 | 100 |
| Total | 280 | 40 | 80 | 400 |

ity to correctly identify the four tissue categories, Benign, InSitu, Invasive, and Normal, across the entire dataset, as defined in Eq. (1) . However, given the clinical significance of misclassification in medical imaging, we further assessed the diagnostic precision and sensitivity of the architectures to ensure a more granular analysis. Precision was used to quantify the exactness of the models, representing the ratio of correctly predicted positive cases to the total predicted positives, as shown in Eq. (2) . Complementary to this, Recall, also known as sensitivity, was calculated to measure the models' ability to detect all relevant positive instances, which is particularly critical in oncology to ensure that malignant tissue types are not overlooked, as formulated in Eq. (3) . Finally, the F1-Score was calculated as the harmonic mean of precision and recall to provide a robust and balanced performance indicator that accounts for both false positives and false negatives, as described in Eq. (4) .

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

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

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

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

## RESULTS AND DISCUSSION

The experimental results obtained from the multi-class classification of breast histopathology images demonstrate a significant variance in performance across the different architectural paradigms. By evaluating traditional CNNs alongside modern ViTs, we assessed not only the predictive accuracy of each model but also their practical feasibility for clinical deployment through the lens of computational complexity. As evidenced by the quantitative outcomes summarized in Table 2, the choice of architecture fundamentally influences the model's ability to distinguish between subtle morphological features in tissue patches.

Among the evaluated models, DenseNet-169 emerged as the superior architecture, achieving the highest accuracy of 0.9625 and an F1-score of 0.9628. Remarkably, DenseNet-169 managed to outperform its counterparts while maintaining the most efficient computational profile, utilizing only 12.49 million parameters and 67.169 GFLOPs. This high performance-to-efficiency ratio is largely attributable to its dense connectivity pattern, which maximizes feature reuse and ensures a robust gradient flow. The confusion matrix for DenseNet-169, illustrated in Figure 2, further confirms its reliability, showing perfect classification (20/20) for the Invasive class and only minor misclassifications in other categories. For instance, only one InSitu sample was incorrectly labeled as Invasive, and a single Normal case was misidentified as Benign, indicating that the model captures the primary pathological indicators with high precision.

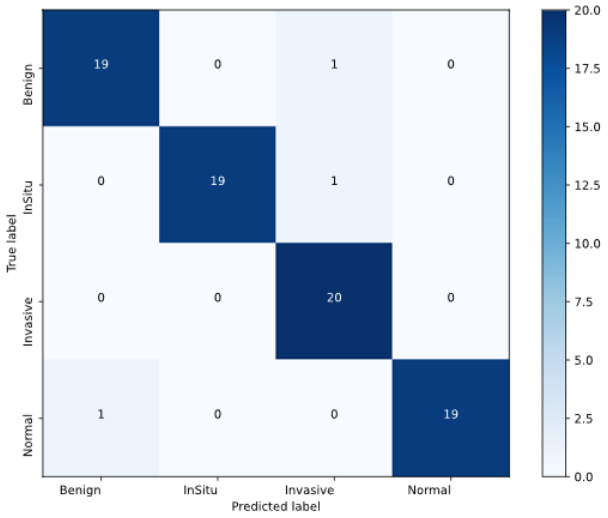



**Figure 2** Confusion Matrix of the DenseNet-169 Model.

In comparison, the DeiT Base model demonstrated competitive diagnostic capabilities with an accuracy of 0.9375, yet it required significantly higher computational resources. With 85.80 million parameters and 336.955 GFLOPs, DeiT is the most demanding architecture in this study, reflecting the inherent trade-offs when using transformer-based models on smaller datasets, despite the use of distillation techniques. Conversely, ResNet-101 performed notably poor in this specific multi-class task, yielding an accuracy of only 0.75. This suggests that for this particular histopathological dataset, the standard residual blocks may not be as effective as the dense connections of DenseNet or the global attention mechanisms of DeiT in resolving the complex cellular overlaps. These findings underscore that while transformers offer powerful global context, optimized CNN architectures like DenseNet-169 currently provide a more balanced solution for high-precision, resource-efficient medical image analysis.

**Table 2** Comparison of Performance Metrics and Computational Complexity of the Models

| Models | Accuracy | Precision | Recall | F1-score | Params | GFLOPs |
|---|---|---|---|---|---|---|
| DeiT Base | 0.9375 | 0.9410 | 0.9375 | 0.9370 | 85.80M | 336.955 |
| DenseNet-169 | 0.9625 | 0.9648 | 0.9625 | 0.9628 | 12.49M | 67.169 |
| ResNet-101 | 0.7500 | 0.7602 | 0.7500 | 0.7487 | 42.51M | 157.288 |

## CONCLUSION

This research has presented a comprehensive comparative analysis of CNN and transformer-based architectures for the multi-class classification of breast cancer histopathology images. By benchmarking DenseNet-169, DeiT Base, and ResNet-101, we have identified that architectural design plays a critical role in capturing the subtle morphological nuances required for accurate tissue differentiation. Our results identify DenseNet-169 as the most effective model, combining exceptional diagnostic accuracy (96.25%) with the most efficient computational profile. While ViTs like DeiT offer promising global contextual awareness, their elevated computational demands and data requirements remain significant barriers compared to the feature-reuse capabilities of dense convolutional networks. These insights underscore the potential for optimized CNNs to be integrated into real-time clinical workflows to alleviate pathologist workload and reduce diagnostic variability. Future work focuses on integrating these models into broader whole-slide imaging pipelines and exploring hybrid architecture to further bridge the gap between local tissue analysis and global architectural interpretation.

### Ethical standard

The authors have no relevant financial or non-financial interests to disclose.

### Availability of data and material

The data that support the findings of this study are available from the corresponding author upon reasonable request.

### Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## LITERATURE CITED


Ahmed, F., 2025 Histovit: Vision transformer for accurate and scalable histopathological cancer diagnosis .

Akbari, Y., F. Abdullakutty, S. Al-Maadeed, A. Bouridane, and R. Hamoudi, 2026 Dynamic model scaling based on segmented tumor size for breast cancer detection. Biomedical Signal Processing and Control **113**: 109118.

Alkhafaji, S. K. D., S. Abdulla, H. A. Marhoon, M. Diykh, M. A. Majed, *et al.*, 2026 Ect-dlm: Deep learning-based empirical curvelet transform approach for thoracic disease diagnosis from x-ray images pp. 37–45.

Dataset, 2025 Bach: Breast cancer histology images.

Gu, Q., N. Prodduturi, and S. N. Hart, 2024 Deep learning in automating breast cancer diagnosis from microscopy images. Proceedings of the 2024 Design of Medical Devices Conference, DMD 2024 .

Hasan, M., M. S. Islam, and M. J. Pathan, 2025 An advanced hybrid deep neural network for precise multiclass classification and detection of cancerous and lymphatic node states. 2025 International Conference on Electrical, Computer and Communication Engineering, ECCE 2025 .

He, K., X. Zhang, S. Ren, and J. Sun, 2025 Deep residual learning for image recognition .

Huang, G., Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, 2025 Densely connected convolutional networks .

Irmak, G. and A. Saygılı, 2025 Deep learning-based histopathological classification of breast tumors: a multi-magnification approach with state-of-the-art models. Signal, Image and Video Processing **19**: 578–.

Jahan, I., M. E. Chowdhury, S. Vranic, R. M. A. Saady, S. Kabir, *et al.*, 2025 Deep learning and vision transformers-based framework for breast cancer and subtype identification. Neural Computing and Applications **37**: 9311–9330.

Kansal, K., S. Kumar, and K. Kansal, 2025 Advances in deep learning techniques for breast cancer classification: A comprehensive review. Archives of Computational Methods in Engineering pp. 1–36.

Krishnappa, S. G. and K. R. U. K. Reddy, 2024 Enhancing histopathology breast cancer detection and classification with the deep ensemble graph network. SN Computer Science **5**: 1–11.

Liu, Y., X. Liu, and Y. Qi, 2024 Adaptive threshold learning in frequency domain for classification of breast cancer histopathological images. International Journal of Intelligent Systems **2024**: 9199410.

Ma, S., H. Du, K. M. Curran, A. Lawlor, and R. Dong, 2024 Adaptive curriculum query strategy for active learning in medical image classification. Lecture Notes in Computer Science **15011 LNCS**: 48–57.

Maurya, R., N. N. Pandey, M. K. Dutta, and M. Karnati, 2024 Fccs-net: Breast cancer classification using multi-level fully convolutional-channel and spatial attention-based transfer learning approach. Biomedical Signal Processing and Control **94**: 106258.

Mumuni, A., F. Mumuni, and N. K. Gerrar, 2024 A survey of synthetic data augmentation methods in machine vision. Machine Intelligence Research 2024 21:5 **21**: 831–869.

Rahaman, M. M., E. K. Millar, and E. Meijering, 2024 Histopathology image classification using supervised contrastive deep learning. Proceedings - International Symposium on Biomedical Imaging .

Rahman, M. A., M. S. H. Khan, Y. Watanobe, J. T. Prioty, T. T. Annita, *et al.*, 2025 Advancements in breast cancer detection: A review of global trends, risk factors, imaging modalities, machine learning, and deep learning approaches. BioMedInformatics 2025, Vol. 5, **5**.

Singh, P., R. Kumar, M. Gupta, and A. J. Obaid, 2024 Transfer learning based breast cancer classification using histopathology images. Proceedings of 2nd International Conference on Advancements in Smart, Secure and Intelligent Computing, ASSIC 2024 .

Singh, S. K. and K. S. Patnaik, 2025 Patho-ai: A perceptive breast cancer identification and classification using deep learning methods integrated with explainable ai. SN Computer Science **6**: 1–17.
Singh, S. K. and K. S. Patnaik, 2026 Mammxai: An xai integrated adaptive multi-model deep learning approach for breast cancer detection using multi-modality images. Biomedical Signal Processing and Control **113**: 109173.
Sumitha, A. and R. S. R. Isaac, 2025 Enhancing breast cancer diagnosis through optimized deep learning and histopathological image analysis. 2025 Global Conference in Emerging Technology, GINOTECH 2025 .
Touvron, H., M. Cord, M. Douze, F. Massa, A. Sablayrolles, *et al.*, 2025 Training data-efficient image transformers & distillation through attention .
Ukwuoma, C. C., D. Cai, E. O. Eziefuna, A. Oluwasanmi, S. F. Abdi, *et al.*, 2025 Enhancing histopathological medical image classification for early cancer diagnosis using deep learning and explainable ai – lime & shap. Biomedical Signal Processing and Control **100**: 107014.
Wang, Z., P. Wang, K. Liu, P. Wang, Y. Fu, *et al.*, 2024 A comprehensive survey on data augmentation .
Yan, Y., R. Lu, J. Sun, J. Zhang, and Q. Zhang, 2025 Breast cancer histopathology image classification using transformer with discrete wavelet transform. Medical Engineering & Physics **138**: 104317.
Yilmaz, M. T., E. Algul, and I. Pacal, 2025 A comparative study of advanced deep learning architectures for breast cancer classification on ultrasound and histological images. Results in Engineering **28**: 107600.
Yu, X., J. Tian, Z. Chen, Y. Meng, and J. Zhang, 2024 Predictive breast cancer diagnosis using ensemble fuzzy model. Image and Vision Computing **148**: 105146.
Zhang, C., P. Chen, and T. Lei, 2025 Category-weight instance fusion learning for unsupervised domain adaptation on breast cancer histopathology images. Biomedical Signal Processing and Control **99**: 106794.

***How to cite this article:*** Eryuksel, M. A., Cakmak, Y., and Pacal, I. A Systematic Benchmark of Advanced Architectures for Automated Multi-Class Classification in Digital Pathology. *ADBA Computer Science*, 3(1), 57-62, 2026.



ADBA Computer Science
Cutting-Edge Scientific Solutions

e-ISSN: 3023-8544
RESEARCH ARTICLE
Vol.3 / No.1 / 2026 / pp.63-71
https://doi.org/10.69882/adba.cs.2026019

# Predictive Modeling for Milk Quality Using Machine Learning and XAI Algorithms

**Zeynep Çetinkaya** [α,1], **Fahrettin Horasan** [α,2], **Hüseyin Aydilek** [β,3] **and Mustafa Yasin Erten** [β,4]
[α]Department of Computer Engineering, Kırıkkale University, Kırıkkale, Türkiye, [β]Department of Electrical & Electronics Engineering, Kırıkkale University, Kırıkkale, Türkiye.

**ABSTRACT** Milk quality assessment is of critical importance for food safety and public health. Traditional milk quality evaluation relies on physicochemical measurements that require expert interpretation and may not directly support rapid or large-scale decision-making, increasing the need for data-driven and automated assessment methods. Although machine learning-based approaches have been widely applied in milk quality classification in recent years, the lack of transparency in model decision mechanisms and insufficient reporting of data preprocessing and data leakage control procedures pose significant limitations in terms of reliability. In this study, the milk quality classification performance of various supervised machine learning algorithms is comparatively evaluated using an open-access milk dataset. Random Forest (RF), k-Nearest Neighbors (KNN), Support Vector Machines (SVM), Extreme Gradient Boosting (XGB), Light Gradient Boosting Machine (LGBM), and Artificial Neural Network (ANN) models are assessed under fair and consistent experimental conditions. The main contribution of this study lies in the application of group-based data partitioning strategies to prevent data leakage, rather than directly removing duplicate or highly similar records from the dataset. This approach prevents data loss and enables a more realistic evaluation of model performance. Furthermore, a targeted and minimalist preprocessing strategy is adopted by applying scaling exclusively to continuous variables. For hyperparameter optimization, Grid Search and Particle Swarm Optimization (PSO) methods are employed; notably, tree-based models optimized using PSO demonstrate more consistent classification performance. To move beyond predictive accuracy, Explainable Artificial Intelligence (XAI) approaches are utilized to enhance the interpretability of model decisions. In this context, SHapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME) methods are applied to analyze the contributions of key features influencing milk quality. Experimental results indicate that physicochemical properties, particularly pH, fat content, and temperature, play a decisive role in milk quality prediction. In conclusion, this study demonstrates that the combined use of machine learning and explainability techniques provides significant contributions in terms of reliability, transparency, and methodological robustness in milk quality classification.

**KEYWORDS**
XAI
Optimization algorithms
Machine learning
Deep learning
Milk quality

## INTRODUCTION

The food industry holds strategic importance on a global scale, both in terms of economic growth and public health. The safety, nutritional value, and sensory acceptability of food products offered on the market directly affect not only consumer satisfaction but also public health and sustainable development. In this context, food quality emerges as a decisive factor in shaping responsible production and consumption practices and plays a critical role in protecting social welfare (Peri 2006).

Within this framework, milk is regarded as a basic and strategic food product due to its high nutritional value and widespread consumption worldwide (Polat *et al.* 2021). However, milk's susceptibility to microbial spoilage and adulteration necessitates the effective and reliable implementation of quality control processes (Azad and Ahmed 2016; Polat *et al.* 2021). Therefore, milk quality assessment is considered a critical step in both safeguarding consumer health and ensuring the reliability of the food supply chain (Murphy *et al.* 2016). Traditional milk quality control methods rely on manual inspections, physical tests, and chemical analyses, and often involve time-consuming, costly, and error-prone procedures (Azad and Ahmed 2016). This situation creates significant constraints in meeting the modern food industry's increasing demands for speed, accuracy, and sustainability.

Recent advances in data science and machine learning have created significant opportunities for the automated, rapid, and reliable assessment of milk quality (Sarveswaran *et al.* 2023; Sunithamani *et al.* 2024). Studies in the literature demonstrate that machine learning-based approaches can determine milk quality with high accuracy while reducing operational costs (Sarveswaran *et al.* 2023; Manisha and Jagadeeshwar 2023). These methods enhance the effectiveness of quality control processes by minimizing human-induced errors. However, most of these high-performance models operate as "black boxes" and fail to provide sufficient explanations of their underlying decision-making mechanisms (Ribeiro *et al.* 2016).

In application areas that directly affect public health, such as food safety, high predictive accuracy alone is insufficient. It is also essential that model decisions are transparent, interpretable, and justifiable (Arrighi *et al.* 2025). Otherwise, the opacity of decision-making processes may lead to trust issues and hinder the adoption of AI-supported systems in real-world applications.

**Manuscript received:** 2 December 2025,
**Revised:** 22 January 2026,
**Accepted:** 25 January 2026.

[1]zeynepcetinkaya@kku.edu.tr (**Corresponding author**)
[2]fhorasan@kku.edu.tr
[3]huseyinaydilek@kku.edu.tr
[4]mustafaerten@kku.edu.tr

In this context, Explainable Artificial Intelligence (XAI) approaches offer an important solution by rendering the decision-making processes of machine learning models more transparent. XAI methods enable the analysis of which variables influence model outputs and to what extent, thereby increasing user trust and ensuring decision verifiability (Ribeiro *et al.* 2016). The integration of XAI into food quality assessment systems allows domain experts to interpret model results more reliably and enhances the practical applicability of these systems (Arrighi *et al.* 2025; Chowdhury *et al.* 2024).

In this study, a comprehensive analysis of milk quality classification was conducted using multiple machine learning algorithms, based on fundamental milk quality parameters reported in the literature. Random Forest (RF), k-Nearest Neighbors (KNN), Support Vector Classifier (SVC), Extreme Gradient Boosting (XGB), Light Gradient Boosting Machine (LGBM), and Artificial Neural Network (ANN) models were evaluated comparatively. Furthermore, the interpretability of model decisions was enhanced using XAI techniques, namely LIME and SHAP, through which critical variables influencing milk quality were examined in detail. In particular, SHAP analyses revealed the contribution of each feature to the model predictions and their relative importance in the decision-making process. The findings of this study are expected to contribute to improving food safety in the dairy industry and to supporting artificial intelligence-driven digital transformation processes. The remainder of this article is organized as follows. Section II reviews related studies on milk quality classification reported in the literature. Section III describes the dataset, data preprocessing procedures, and machine learning methods employed. Section IV presents the experimental results and comparative model performance analyses. Section V provides a comparative discussion of the proposed approach with existing studies. Finally, the concluding section summarizes the main findings and offers recommendations for future research.

## LITERATURE REVIEW

The rapid, objective, and reproducible assessment of milk quality is of critical importance for food safety, consumer health, and quality assurance in the dairy industry. Traditional chemical and microbiological analyses are often time-consuming, costly, and highly dependent on operator expertise, which limits their effectiveness in large-scale and real-time quality control processes. Consequently, a growing body of research has emerged in recent years focusing on machine learning, deep learning, explainable artificial intelligence (XAI), Internet of Things (IoT), and blockchain-based approaches for the classification and monitoring of milk quality.

In the literature, studies employing the milk quality grading dataset published on the Kaggle platform have attracted particular attention. chaudhari2025machine compared various machine learning algorithms, including Decision Trees, Random Forests, and Support Vector Machines, to classify milk quality and evaluated model performance using standard classification metrics such as accuracy, precision, recall, and F1 score. Similarly, Bhavsar *et al.* (2023) conducted a detailed analysis of Random Forest and SVM models on the same dataset and demonstrated that Random Forest achieved superior accuracy and generalization performance following appropriate preprocessing and label encoding. Another study Samad *et al.* (2024) compared KNN with its enhanced variant, DW-KNN, for milk quality detection and reported that the proposed method provided improved classification capability. Likewise, Shahzad *et al.* (2025), as well as Çelik (2022), performed accuracy-oriented evaluations by comparing multiple classification algorithms on the Kaggle milk dataset. A common characteristic of these studies is the use of a relatively small but well-defined dataset to demonstrate that milk quality can be automatically classified into low, medium, and high categories.

More recent studies have shifted their focus beyond classification performance to the interpretability of model decision-making processes. Çetintav and Yalçın (2025) performed milk quality classification using Random Forest and HistGradientBoost models on the Kaggle dataset and subsequently integrated XAI techniques such as LIME and Permutation Feature Importance to analyze the contributions of key features, including pH, temperature, and taste, both globally and at the instance level. This approach not only delivered high predictive performance but also provided an explainable framework that transparently revealed the factors underlying milk quality decisions.

At a more advanced methodological level, Kurtanjek (2024) addressed the limitations of correlation-based models by employing causal artificial intelligence approaches. Using Bayesian networks and structural causal modeling, the study investigated causal relationships among factors influencing milk quality, thereby contributing not only to predictive accuracy but also to a deeper understanding of the underlying mechanisms driving quality outcomes. Veena and Poovammal (2025) proposed a hybrid approach combining Principal Component Analysis (PCA) and decision trees with robust scaling to mitigate the effects of outliers. In another study (Tolba *et al.* 2024), a deep learning model integrating GRU and ResNet architectures was evaluated on a milk quality dataset, demonstrating the potential of advanced neural network-based approaches.

Beyond chemical and physicochemical parameters, several studies have addressed milk quality from supply chain and traceability perspectives. Manisha and Jagadeeshwar (2023) introduced a blockchain-supported traceability framework that integrates IoT, blockchain, and deep learning components to ensure transparent and immutable tracking of quality and safety information throughout the food supply chain. Similarly, Goyal *et al.* (2024) developed an AI- and IoT-based multi-sensor system for detecting and classifying milk adulteration, a critical challenge in milk quality assurance; SHAP was employed to enhance the interpretability of model decisions.

Spectroscopic and image-based techniques have also played a significant role in milk quality assessment. One study Thanasirikul *et al.* (2023) utilized color sensor data derived from the rezazurin test to rapidly classify raw milk quality using SVM. Tahtali (2020) examined milk quality based on somatic cell count and compositional parameters in raw milk samples, analyzing the impact of increasing somatic cell levels from a machine learning perspective. Neto *et al.* (2019) achieved high accuracy in detecting milk adulteration and contamination by combining Fourier-transform infrared (FTIR) spectroscopy data with a customized convolutional neural network architecture. Mhapsekar *et al.* (2025) provided a comprehensive review of studies published between 2012 and 2023, comparing machine learning and deep learning approaches for milk quality assessment in terms of data types, methodologies, performance metrics, and emerging research trends.

Despite the extensive literature, studies focusing specifically on the Kaggle milk quality grading dataset that systematically define preprocessing procedures, evaluate multiple machine learning algorithms under fair and consistent conditions, and comprehensively analyze feature importance and decision-making processes within an XAI framework remain limited. This gap highlights the need for more robust and transparent methodological approaches

that achieve both high classification accuracy and strong interpretability in milk quality assessment.

## MATERIAL AND METHOD

The methodology adopted in this study is based on the systematic execution of the fundamental stages of the experimental workflow. The process begins with data acquisition, followed by data preprocessing, exploratory analysis of data distributions, and partitioning of the dataset into training and test subsets. After model training and hyperparameter optimization using the training data, the models are evaluated and compared on the test data to identify those with the highest predictive performance. In the final stage, the decision-making mechanisms of the selected models are analyzed using Explainable Artificial Intelligence (XAI) techniques based on LIME and SHAP, and the key features influencing model predictions are examined in detail.

### Data Set

In this study, the Milk Quality Dataset published on the Kaggle open data platform was used to develop machine learning models for milk quality classification (Shrijayan (cpluzshrijayan) n.d.). The dataset comprises a total of 1,059 samples, each representing a milk sample described by physical and sensory quality measurements. The data were collected through manual observations by researchers and shared under an open data license ("EU ODP Legal Notice") (Çetintav and Yalçın 2025; Manisha and Jagadeeshwar 2023). The dataset includes seven independent variables that characterize the physical and sensory properties of milk, along with a target variable (Grade) representing overall milk quality:

- pH: Acidity level of milk (numerical; range: 3.0–9.5).
- Temperature: Sample temperature in °C (numerical; range: 34–90).
- Taste: Indicator of taste quality (0 = poor, 1 = good).
- Odor: Indicator of odor quality (0 = poor, 1 = good).
- Fat: Presence of fat content (0 = absent, 1 = present).
- Turbidity: Turbidity level (0 = low, 1 = high).
- Color: Color intensity of the sample (numerical; range: 240–255).

The target variable Grade is categorized into three classes based on sensory and physical criteria: Low (Bad), Medium (Moderate), and High (Good). Although the dataset primarily consists of binary variables, the inclusion of continuous features such as pH, temperature, and color enables the application of both classical machine learning algorithms and explainable artificial intelligence (XAI) techniques [3]. Owing to its open-access nature and the inclusion of parameters widely regarded as critical for milk quality assessment, this dataset provides a suitable and well-established basis for machine learning-based classification and model interpretability studies.

### Data Preparation

In this study, a systematic preprocessing pipeline was applied to the dataset to enhance the reliability and generalizability of the machine learning models. In the initial stage, the dataset was inspected for missing values and data type inconsistencies. No missing observations were identified, and all variables were confirmed to have data types suitable for analysis. A detailed examination revealed that a substantial proportion of the samples in the dataset consisted of duplicate records. When these duplicate records were removed, the number of unique samples decreased markedly, which significantly constrained the statistical representativeness of the dataset and the learning capacity of the models. Therefore, to avoid data loss, duplicate records were retained; however, to prevent data leakage, group-based splitting strategies were employed during the model evaluation phase. This strategy ensured that duplicate records corresponding to the same original sample did not appear simultaneously in both the training and test sets. During the modeling process, label encoding was applied to the target variable, Grade. As most of the independent variables are binary, feature scaling was applied exclusively to the continuous variables—pH, Temperature, and Color. These variables were normalized to the [0,1] range using Min–Max normalization. The Min–Max scaling procedure is defined as follows:

$$X_{\text{scaled}} = \frac{X - \min(X)}{\max(X) - \min(X)} \tag{1}$$

Here, Xscaled denotes the scaled (normalized) value, X represents the original value, min(X) and max(X) correspond to the minimum and maximum values of the feature across the entire dataset, respectively. This normalization strategy is adopted to prevent differences in feature scales from adversely affecting model performance, particularly for distance-based algorithms such as k-nearest neighbors, support vector machines, and logistic regression.

An examination of class proportions in the dataset revealed no significant class imbalance. The class distributions in the training and test sets are illustrated in Fig. 1. Accordingly, neither oversampling nor undersampling techniques were applied, in order to avoid introducing bias into the model learning process through artificial data manipulation. As a result of these preprocessing steps, the dataset was rendered suitable for training machine learning models and for conducting explainability analyses.

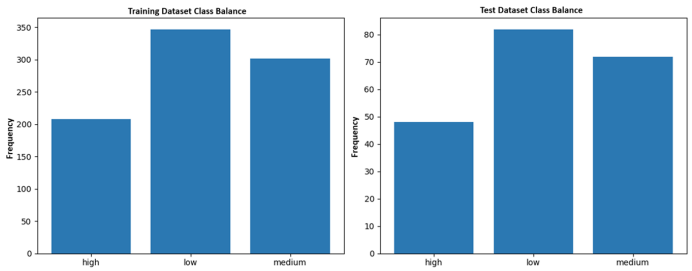



**Figure 1** Distribution of class labels in the training and test datasets

### Models Preparation

In this study, machine learning models with diverse structural characteristics were evaluated for milk quality classification. Within the scope of the comparative analysis, RF, KNN, SVC, XGB, LGBM, and ANN models were employed. Following an initial assessment using baseline hyperparameter settings, hyperparameter optimization was conducted to enhance model performance. GridSearchCV was adopted as a conventional baseline approach for optimization within limited and discrete hyperparameter spaces. However, due to the high computational cost and the exhaustive nature of grid-based search particularly in continuous and high-dimensional parameter spaces a more flexible and computationally efficient optimization strategy was required.

Accordingly, Particle Swarm Optimization (PSO)-based hyperparameter tuning was applied, particularly to tree-based models

(RF, XGB, and LGBM) and ANN architectures. PSO is a population-based heuristic optimization algorithm that facilitates rapid convergence toward global optima in complex and continuous search spaces. By iteratively updating candidate solutions (particles) based on both individual best positions and the global best solution, PSO enables a more efficient exploration of the hyperparameter space compared to grid-based methods. During the hyperparameter optimization process, both accuracy and macro-averaged F1 score were jointly considered as performance metrics. While accuracy reflects overall classification performance, the macro-F1 score assigns equal importance to each class, thereby mitigating potential biases arising from class distribution differences. All optimization procedures were conducted in accordance with the group-based data splitting strategy implemented to prevent data leakage caused by repeated samples. The optimal hyperparameter configurations obtained for each model are reported in Table 1.

### Evaluation Criteria

To comprehensively and reliably evaluate the performance of the classification models developed in this study, four widely used performance metrics were considered: accuracy, precision, recall, and F1 score. Accuracy represents the overall performance of a model and is defined as the ratio of correctly classified instances to the total number of instances. However, accuracy alone may be insufficient in scenarios involving class imbalance or when certain classes are of greater importance. Precision evaluates the extent of false positive predictions by indicating the proportion of instances predicted as positive that are actually correct. Recall, in contrast, captures the impact of false negatives by measuring the proportion of true positive instances that are correctly identified. The F1 score, which provides a balance between precision and recall, is defined as the harmonic mean of these two metrics and enables a more robust comparison of model performance across classes (Chaudhari *et al.* 2025; Horasan *et al.* 2019). The mathematical definitions of these performance metrics are provided in Equations (2)–(5).

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ Instances} \tag{2}$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{3}$$

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{4}$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5}$$

In addition to these quantitative metrics, a confusion matrix was employed to analyze model performance on a per-class basis in greater detail. The confusion matrix offers a visual representation of correct and incorrect predictions for each class, thereby clearly illustrating the strengths and weaknesses of the models with respect to individual class predictions.

### Explainable Artificail Intelligent

In this study, the classification model that achieved the highest performance during the evaluation was selected for the analysis of the explainability. To enhance the transparency and interpretability of the decision-making mechanism of the selected model, two Explainable Artificial Intelligence (XAI) techniques, LIME and SHAP were jointly employed.

Due to its model-agnostic nature, the LIME method constructs a local linear approximation around a specific instance, thereby revealing which features influence individual predictions and in which direction. This capability provides a substantial advantage, particularly in interpreting and justifying instance-level predictions and in understanding the behavior of the model in specific samples (Ribeiro *et al.* 2016; Chowdhury *et al.* 2024).

The SHAP method, in contrast, leverages Shapley values derived from cooperative game theory to quantitatively estimate the contribution of each feature to the model output, offering explanations at the global level. This enables a systematic assessment of the variables that shape the overall decision structure of the model and their relative importance throughout the dataset (Sermmany *et al.* 2024; Shapley 1953).

By combining LIME and SHAP, it becomes possible to explain both local (instance-level) and global (model-level) behaviors of the selected classifier. Consequently, instead of simply reporting high performance metrics, the decision-making processes of the model are examined within a comprehensive and scientifically grounded explainability framework, thus substantially strengthening the interpretability of the proposed approach.

## EXPERIMENTAL RESULTS

### Models Performance

In this study, various machine learning models were evaluated for milk quality classification, and their performance was compared using accuracy, macro-precision, macro-recall, and macro-F1 score metrics. The obtained results are summarized in Table 2.

An examination of Table II shows that the XGB and LGBM models optimized using PSO achieved the highest performance across all evaluation metrics. In particular, the XGB(PSO) model emerged as a representative high-performing model, achieving a test accuracy of 89.1 % and a Macro-F1 score of 0.8777. These findings indicate that boosting-based methods are capable of effectively modeling the complex and non-linear relationships inherent in milk quality data.

Although RF(PSO) and ANN(PSO) demonstrated moderate performance, the LR, KNN, and SVC models optimized via Grid Search showed notably lower results in comparison. The relatively low Macro-F1 scores of these models suggest limitations in maintaining balanced performance across classes. Overall, the results highlight that PSO-based hyperparameter optimization substantially enhances classification performance, particularly for ensemble and boosting-based models.

Fig. 2 illustrates the confusion matrices obtained on the test dataset for the PSO-optimized XGB, RF, and ANN models. The XGB(PSO) model correctly classifies the high and low classes without error, whereas a limited number of medium class instances are misclassified as high. In contrast, the RF(PSO) and ANN(PSO) models exhibit a more pronounced confusion between the medium and high classes. These observations further support the superior inter-class discrimination capability of the XGB model and are consistent with the higher Macro-F1 score reported in Table 2.

### XAI Results

To enhance the transparency of the proposed classification framework, explainable artificial intelligence (XAI) techniques were employed to analyze both global and local decision mechanisms of the optimized XGBoost model. In this context, SHAP was adopted as the primary explanation method, while LIME was used to provide complementary local verification.

**Table 1** est Hyperparameters and Optımızatıon Methods Selected for the Compared Models

| Model | Optimization | Best hyperparameters |
|---|---|---|
| Logistic Regression | Grid Search | C=100, solver=lbfgs |
| KNN | Grid Search | n_neighbors=3, p=2, weights=distance |
| SVM (RBF) | Grid Search | C=100, gamma=scale, kernel=rbf |
| Random Forest | PSO | n_estimators=353, max_depth=14, min_samples_split=2, min_samples_leaf=1 |
| XGBoost | PSO | n_estimators=588, max_depth=9, learning_rate≈0.243, subsample≈0.675, colsample_bytree≈0.957, reg_lambda≈2.70 |
| LightGBM | PSO | n_estimators=231, num_leaves=104, learning_rate≈0.295, min_child_samples=31, subsample≈0.657, colsample_bytree≈0.913 |
| ANN (MLP) | PSO | hidden_layer_sizes=(87,), alpha≈ $6.1 \times 10^{-4}$, learning_rate_init≈0.0118 |

**Table 2** Performance Table

| Model | Test_Accuracy | Test_MacroPrecision | Test_MacroRecall | Test_MacroF1 |
|---|---|---|---|---|
| XGB(PSO) | 0.891089 | 0.895238 | 0.898148 | 0.877744 |
| LGBM(PSO) | 0.891089 | 0.895238 | 0.898148 | 0.877744 |
| RF(PSO) | 0.811881 | 0.852713 | 0.845528 | 0.804944 |
| ANN(PSO) | 0.782178 | 0.834028 | 0.807588 | 0.781656 |
| LR(Grid) | 0.594059 | 0.766026 | 0.653117 | 0.596633 |
| KNN(Grid) | 0.594059 | 0.766026 | 0.653117 | 0.596633 |
| SVC(Grid) | 0.594059 | 0.691135 | 0.653117 | 0.581848 |

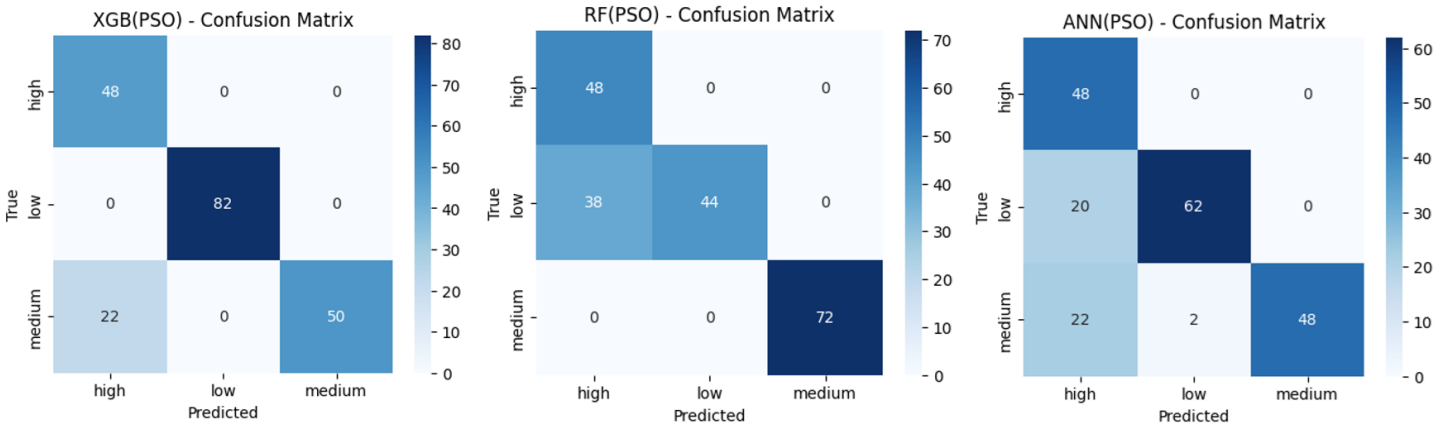



**Figure 2** Confusion matrices obtained on the test dataset for the PSO-optimized XGB, RF, and ANN models.

Figure 3 presents the global feature importance derived from mean absolute SHAP values. The results indicate that pH, Fat, and Temperature are the most influential variables in the model's decision process, followed by Odor, Colour, Turbidity, and Taste. This ranking demonstrates that physicochemical properties of milk, particularly acidity level and fat content, play a dominant role in quality grade discrimination. Compared to discrete sensory-related attributes, continuous physicochemical features exhibit

stronger and more stable contributions.

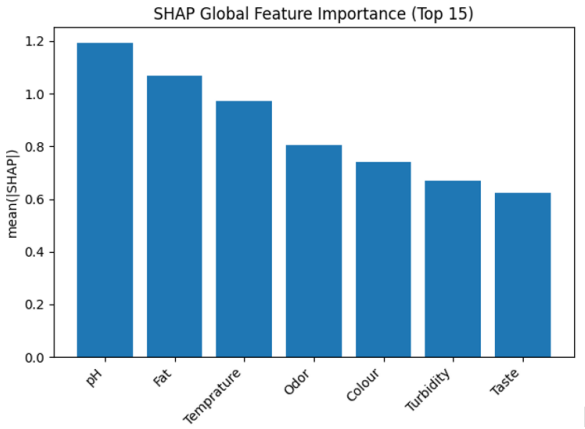



**Figure 3** SHAP Global Feature Importance Based on Mean Absolute SHAP Values.

While global importance rankings reveal which features matter most, they do not capture how feature values influence class predictions. To address this, Figure 4 illustrates the SHAP summary (beeswarm) plot for the most frequently observed class. In this visualization, each point represents an individual sample, colored according to its feature value. Positive SHAP values indicate an increased likelihood of the target class, whereas negative values reduce the corresponding class score. The plot reveals that pH, Fat, and Temperature not only have high importance but also exhibit bidirectional effects, depending on their observed values. This behavior highlights their role in defining class boundaries rather than acting as monotonic predictors. These findings underline that milk quality assessment is governed by complex feature interactions, where identical variables may support different class outcomes under varying conditions.

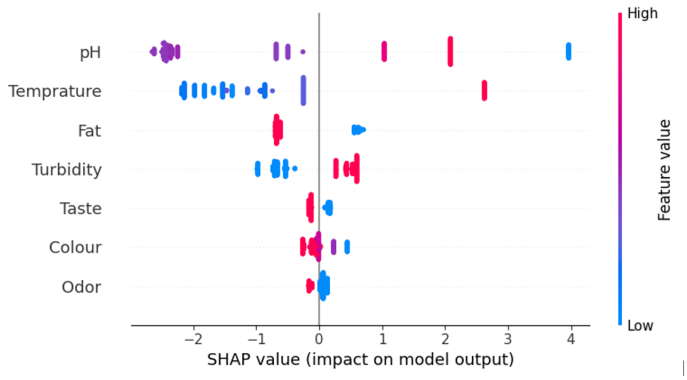



**Figure 4** SHAP Summary (Beeswarm) Plot for the Most Frequent Class.

To further investigate the model's decision-making process at the instance level, a local SHAP analysis was conducted on a misclassified test sample. Figure 5 shows the SHAP waterfall plot for an instance whose true label was medium but was predicted as high. The visualization reveals that Fat, Colour, Taste, and Temperature contributed positively toward the high class prediction, while pH and Odor exerted negative influence. Despite the presence of counteracting signals, the cumulative positive contributions outweighed the negative effects, resulting in an incorrect classification.

This example demonstrates how borderline samples located near class boundaries may be sensitive to competing feature contributions, leading to misclassification even in high-performing models.

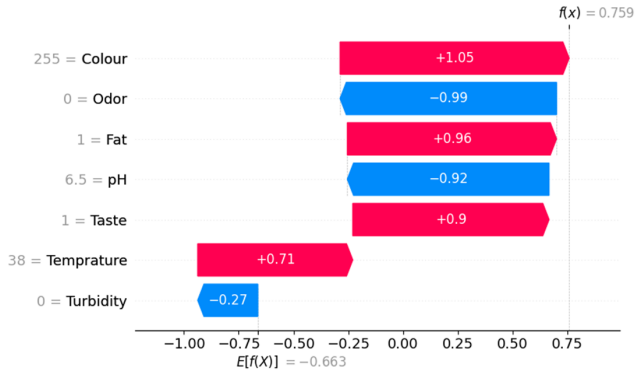



**Figure 5** Local SHAP Explanation of a Misclassified Test Instance.

To validate the reliability of the SHAP-based interpretations, a LIME local explanation was generated for the same test instance. Figure 6 presents the LIME explanation for the predicted high class. Consistent with the SHAP analysis, LIME identified Fat, Colour, Taste, and Temperature as positive contributors to the high class, while pH and Odor acted as negative factors. The agreement between SHAP and LIME reinforces the robustness of the observed explanations and confirms that the model's local decision logic is not dependent on a single interpretability method.

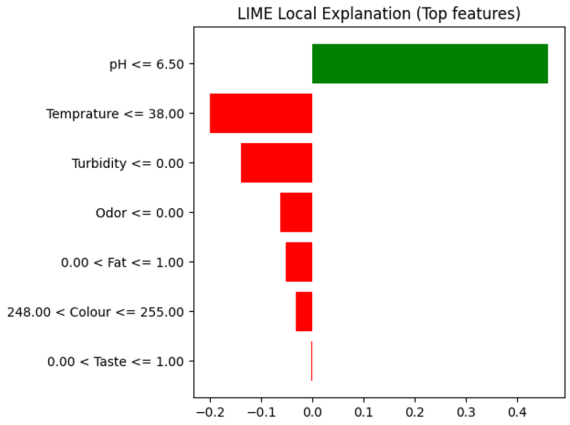



**Figure 6** LIME Local Explanation for the Predicted High Class.

Overall, the combined global and local XAI analyses demonstrate that the proposed model relies on meaningful and domain-consistent features when assessing milk quality. The use of group-based data splitting ensures that these explanations are not affected by data leakage, thereby increasing their reliability. Moreover, the consistency between SHAP and LIME interpretations supports the transparency and trustworthiness of the model, particularly in borderline cases where classification uncertainty is inherently higher.

## COMPARISON WITH RELATED STUDIES

When compared with existing studies in the literature on milk quality classification, the approach proposed in this study exhibits notable differences, particularly with respect to data preprocessing

strategies, and highlights several common methodological gaps. The vast majority of the reviewed studies rely on the same publicly available dataset obtained from Kaggle (1,059 samples with 7 features) and adopt similar preprocessing assumptions, largely due to the absence of missing values in the dataset. However, the presence of duplicate records an issue that is critical for data integrity and the reliability of model evaluation and the manner in which such records are handled are not explicitly reported in most studies (Bhavsar *et al.* 2023; Çetintav and Yalçın 2025; Shahzad *et al.* 2025; Çelik 2022; Mu *et al.* 2020; Samad *et al.* 2024; Kurtanjek 2024). Only Study 1 reports the identification and direct removal of duplicate samples from the dataset.

The analyses conducted in the present study revealed a substantial number of duplicate records within the dataset. It was determined that directly eliminating these records could significantly reduce the representativeness of the data and adversely affect the learning capacity of the models. To address this issue, group-based data partitioning strategies were employed to prevent data leakage while preserving the full set of available samples. This strategy ensured that duplicate instances originating from the same source did not appear simultaneously in both the training and test sets, thereby enabling a more realistic and reliable evaluation of model performance. To the best of our knowledge, none of the existing studies in the literature explicitly address duplicate data management at this level.

With respect to feature scaling, Min–Max normalization is commonly adopted in the literature (Bhavsar *et al.* 2023; Çetintav and Yalçın 2025; Shahzad *et al.* 2025; Samad *et al.* 2024), while StandardScaler is preferred in (Chaudhari *et al.* 2025; Tolba *et al.* 2024), and (Kumari *et al.* 2023), Robust Scaling is applied only in (Veena and Poovammal 2025). In contrast, the present study adopts a targeted preprocessing strategy in which scaling is not applied to binary variables, and Min–Max normalization is used exclusively for continuous features such as pH, temperature, and color. This selective approach aims to improve the performance of distance-based algorithms while avoiding unnecessary transformations of discrete sensory-related attributes.

Different strategies have also been reported in the literature to address potential class imbalance. While SMOTE is employed in (Tolba *et al.* 2024), balance is achieved through strategic sample selection in (Çelik 2022), and (Çetintav and Yalçın 2025) reports that the dataset is inherently balanced. In the distribution analyses conducted in this study, no significant class imbalance was detected. Consequently, oversampling or undersampling techniques were not applied in order to avoid introducing additional bias through artificial manipulation of the data distribution.

A systematic comparison of the proposed methodology with related studies is provided in Table 3. As summarized in the table, most existing studies do not explicitly report how duplicate records are handled, nor do they apply selective scaling strategies or group-based data partitioning. In contrast, the proposed approach integrates these steps in a unified preprocessing framework, highlighting its methodological distinctiveness.

When the accuracy comparisons presented in Table 4 are examined, it becomes evident that many of the high performance values reported in the literature are obtained without explicitly addressing duplicate data management or potential data leakage risks. In contrast, the cautious and controlled preprocessing strategy adopted in this study results in lower accuracy values for some models; however, these results are considered to more accurately reflect real-world generalization performance, as they are obtained under strict data leakage control conditions.

Overall, this study introduces a minimalist and reliability-oriented preprocessing philosophy by prioritizing duplicate-aware data partitioning, avoiding unnecessary scaling operations, and refraining from artificial data balancing techniques. Future studies may further investigate the origins of duplicate samples (e.g., repeated measurements or data acquisition artifacts) and systematically evaluate the impact of different normalization strategies on model stability and robustness.

## CONCLUSION

This study comparatively evaluates the classification performance of various supervised machine learning algorithms using an open-access milk quality dataset. The primary objective of the study is to move beyond approaches that focus solely on high accuracy values and to present a holistic methodological framework that jointly considers the reliability of data preprocessing, the validity of model evaluation, and the interpretability of decision-making processes. The findings indicate that, particularly in datasets containing repeated samples derived from the same source, data leakage can artificially inflate model performance. Accordingly, instead of directly removing duplicate records, group-based data partitioning strategies were employed to prevent data loss while preserving the models' ability to generalize to unseen data. Through this approach, models were evaluated under more realistic conditions, and attention was drawn to an important methodological issue that is frequently overlooked in the literature.

During the data preprocessing stage, scaling was applied exclusively to the necessary continuous variables, and artificial resampling techniques were deliberately avoided for a dataset that exhibited no class imbalance. This targeted and controlled data preparation strategy highlights that, especially for datasets with a high proportion of duplicate samples, detailed data analysis and informed preprocessing decisions are as critical as the choice of learning algorithm itself. The modeling results demonstrate that tree-based and boosting-based models optimized using Particle Swarm Optimization (PSO) achieve more balanced and consistent performance in milk quality classification. These findings suggest that PSO-based hyperparameter optimization provides a more efficient and effective search process in high-dimensional and continuous parameter spaces compared to traditional grid-based approaches.

To move beyond predictive performance, explainable artificial intelligence (XAI) techniques were employed to analyze the decision-making processes of the models. Using LIME and SHAP, physicochemical attributes such as pH, fat content, and temperature were identified as key determinants in milk quality prediction. This analysis enables model outputs to be evaluated not only through numerical performance metrics but also in an interpretable and well-justified manner. In conclusion, this study presents a reliable, transparent, and methodologically sound machine learning framework for milk quality classification by jointly addressing duplicate data management, data leakage prevention, targeted data preprocessing, hyperparameter optimization, and explainability analysis.

Future studies aim to further investigate the impact of explainable AI analyses across different model architectures and data partitioning strategies, as well as to evaluate the contribution of duplicate-aware data management approaches to generalization performance on larger and more diverse milk quality datasets. In addition, future work may explore the integration of domain knowledge and sensor-level metadata to further enhance interpretability and robustness.

**Table 3** Comparison of the Proposed Approach with Related Studies in the Literature

| Articles | Missing Value Check | Scaling Method | Duplicate Handling | Dimension Reduction | Class Imbalance Analysis |
|---|---|---|---|---|---|
| (Chaudhari *et al.* 2025) | + | StandardScaler | +(removed, impact of data reduction not reported) | - | - |
| (Bhavsar *et al.* 2023) | + | Min–Max | - | - | - |
| (Çetintav and Yalçın 2025) | + | Min–Max | - | - | +(balanced) |
| (Samad *et al.* 2024) | + | Min–Max | - | - | +(balanced) |
| (Veena and Poovammal 2025) | + | Robust Scaling | - | + (PCA) | - |
| (Tolba *et al.* 2024) | + | StandardScaler | - | - | -(imbalanced – SMOTE) |
| (Kumari *et al.* 2023) | + | StandardScaler | +(no duplicate values present) | + (PCA) | - |
| **Ours** | + | Min–Max(continuous only) | +(group-based, leakage-aware) | - | +(balanced, no resampling) |

**Table 4** Accuracy Comparison with Related Studies in the Literature

| Model \ Articles | (Chaudhari *et al.* 2025) | Bhavsar *et al.* (2023) | (Çetintav and Yalçın 2025) | (Samad *et al.* 2024) | (Veena and Poovammal 2025) | (Tolba *et al.* 2024) | (Kumari *et al.* 2023) | **Ours** |
|---|---|---|---|---|---|---|---|---|
| Logistic Regression | 0.36 | - | - | - | 0.60 | - | 0.8490 | 0.5941 |
| Random Forest | 0.99 | 0.92 | 0.995 | - | 0.75 | 0.914 | 0.9968 | 0.8119 |
| SVM | - | 0.57 | 0.566 | - | 0.75 | 0.868 | 0.9528 | 0.5941 |
| KNN | 0.98 | - | 0.985 | 0.985 | - | - | 0.9968 | 0.5941 |
| ANN / MLP | 0.45 | - | - | - | - | 0.933 | - | 0.7822 |
| XGBoost | - | - | 0.973 | - | - | - | - | 0.8911 |
| LightGBM | - | - | 0.970 | - | - | - | - | 0.8911 |

### Ethical standard

The authors have no relevant financial or non-financial interests to disclose.

### Availability of data and material

The data that support the findings of this study are available from the corresponding author upon reasonable request.

### Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## LITERATURE CITED


Arrighi, L., I. A. de Moraes, M. Zullich, M. Simonato, D. F. Barbin, *et al.*, 2025 Explainable artificial intelligence techniques for interpretation of food datasets: a review. arXiv preprint arXiv:2504.10527 .

Azad, T. and S. Ahmed, 2016 Common milk adulteration and their detection techniques. International Journal of Food Contamination **3**: 22.

Bhavsar, D., Y. Jobanputra, N. K. Swain, and D. Swain, 2023 Milk quality prediction using machine learning. EAI Endorsed Transactions on Internet of Things **10**.

Çelik, A., 2022 Using machine learning algorithms to detect milk quality. Eurasian Journal of Food Science and Technology **6**: 76–87.

Çetintav, B. and A. Yalçın, 2025 Explainable machine learning framework for milk quality grading. Kocatepe Veterinary Journal **18**: 227–235.

Chaudhari, A., R. Mane, A. Khot, A. Kadam, and N. Rajam, 2025 Machine learning-based classification for milk quality assessment. In *2025 3rd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, pp. 1441–1446, IEEE.

Chowdhury, R., R. Das, F. B. F. Ananna, A. Saha, S. Nawar, *et al.*, 2024 Unveiling predictive factors in apple quality: Leveraging

lime, shap, and the synergy of machine learning models and artificial neural networks. In *2024 6th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT)*, pp. 1026–1031, IEEE.

Goyal, K., P. Kumar, and K. Verma, 2024 Xai-empowered iot multisensor system for real-time milk adulteration detection. Food Control **164**: 110495.

Horasan, F., H. Erbay, F. Varçın, and E. Deniz, 2019 Alternate low-rank matrix approximation in latent semantic analysis. Scientific Programming **2019**: 1095643.

Kumari, S., M. K. Gourisaria, H. Das, and D. Banik, 2023 Deep learning based approach for milk quality prediction. In *2023 11th International conference on emerging trends in engineering & technology-signal and information processing (ICETET-SIP)*, pp. 1–6, IEEE.

Kurtanjek, Ž., 2024 Causal artificial intelligence models of food quality data. Food Technology and Biotechnology **62**: 102–109.

Manisha, N. and M. Jagadeeshwar, 2023 Bc driven iot-based food quality traceability system for dairy product using deep learning model. High-Confidence Computing **3**: 100121.

Mhapsekar, R., D. Kilbane, S. Davy, L. Abraham, M. Fenelon, *et al.*, 2025 A systematic review of the internet of things and artificial intelligence applications in milk quality monitoring and analysis. International Journal of Dairy Technology **78**: e70049.

Mu, F., Y. Gu, J. Zhang, and L. Zhang, 2020 Milk source identification and milk quality estimation using an electronic nose and machine learning techniques. Sensors **20**: 4238.

Murphy, S. C., N. H. Martin, D. M. Barbano, and M. Wiedmann, 2016 Influence of raw milk quality on processed dairy products: How do raw milk quality test results relate to product quality and yield? Journal of Dairy Science **99**: 10128–10149.

Neto, H. A., W. L. Tavares, D. C. Ribeiro, R. C. Alves, L. M. Fonseca, *et al.*, 2019 On the utilization of deep and ensemble learning to detect milk adulteration. BioData Mining **12**: 13.

Peri, C., 2006 The universe of food quality. Food quality and preference **17**: 3–8.

Polat, O., S. G. Akçok, M. A. Akbay, D. Topaloğlu, S. Arslan, *et al.*, 2021 Classification of raw cow milk using information fusion framework. Journal of Food Measurement and Characterization **15**: 5113–5130.

Ribeiro, M. T., S. Singh, and C. Guestrin, 2016 " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144.

Samad, A., S. Taze, and M. K. Uçar, 2024 Enhancing milk quality detection with machine learning: A comparative analysis of knn and distance-weighted knn algorithms. Int. J. Innov. Sci. Res. Technol **9**: 2021–2029.

Sarveswaran, S., S. Jha, B. Soundarya, *et al.*, 2023 Milksafe: a hardware-enabled milk quality prediction using machine learning. In *2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN)*, pp. 1–6, IEEE.

Sermmany, K., P. Wanjantuk, and W. Leelapatra, 2024 Utilizing explainable artificial intelligence (xai) to identify determinants of coffee quality. In *2024 21st International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pp. 696–703, IEEE.

Shahzad, A., S. Javaid, and Z. Alamsyah, 2025 Milk quality detection using machine learning. Engineering Proceedings **107**: 119.

Shapley, L., 1953 Stochastic gamesproceedings of the national academy of sciences of the usa 39, 1095–1100 (chapter 1 in this volume). MathSciNet zbMATH .

Shrijayan (cpluzshrijayan), n.d. Milk Quality Prediction Dataset. https://www.kaggle.com/datasets/cpluzshrijayan/milkquality, Accessed: 2026-01-25.

Sunithamani, S., D. Muralidhar, G. Anne, and C. N. Sruthi, 2024 Milk quality prediction using machine learning integrated with arduino. In *2024 10th International Conference on Communication and Signal Processing (ICCSP)*, pp. 1268–1273, IEEE.

Tahtali, Y., 2020 Classification of raw milk composition and somatic cell count in water buffaloes with support vector machines. Kafkas Üniversitesi Veteriner Fakültesi Dergisi **26**.

Thanasirikul, C., A. Patumvan, D. Lipsky, S. Bovonsombut, P. Singjai, *et al.*, 2023 Rapid assessment and prediction of microbiological quality of raw milk using machine learning based on rgb-colourimetric resazurin assay. International Dairy Journal **146**: 105750.

Tolba, A., N. Mostafa, A. Mohamed, and K. Sallam, 2024 Hybrid deep learning approach for milk quality prediction. Precis. Livest. J. **1**: 1–13.

Veena, V. and E. Poovammal, 2025 An improved multi classification of milk quality using machine learning. In *2025 2nd International Conference on Trends in Engineering Systems and Technologies (ICTEST)*, volume 1, pp. 1–6, IEEE.

***How to cite this article:*** Çetinkaya, Z., Horasan, F., Aydilek, H., and Erten, M. Y. Predictive Modeling for Milk Quality Using Machine Learning and XAI Algorithms. *ADBA Computer Science*, 3(1), 63-71, 2026.