Emerging Trends in Engineering and Sustainability (ETES) Volume 02 (No. 02), 2026, pp. 68-76

Emerging Trends in Engineering and Sustainability (ETES)

https://etes.alnaji-uni.edu.iq

https://doi.org/10.65505/ETES-26-0012

ISSN 3105-1723 (Print)
ISSN 3105-1731 (Online)

AI-Augmented Networked Control Systems Subject to Network Latency and Packet Loss

Maher Faik Esmaile[1*], Huda Mohammed Akar[2], Ghassan Salem[3]

[1] Department of AI Engineering, College of Engineering, Al-Naji University, Baghdad, Iraq.
[2] Department of Cyber Security, College of Engineering, Al-Naji University, Baghdad, Iraq.
[3] Department of Computer Engineering, College of Engineering, Al-Naji University, Baghdad, Iraq.

Article Info

Received: 13 June 2026
Revised: 27 June 2026
Accepted: 28 June 2026
Available online: 30 June 2026

Keywords:

Networked Control Systems; Communication Delay; Packet Loss; K-Nearest Neighbors; Random Forest; Robust Control; Machine Learning in Control.

ABSTRACT

Networked Control Systems (NCSs) find widespread implementation in industrial systems. However, unreliable networks with time-varying delays and packet losses can hurt system stability, reliability, and efficiency. In this work, we propose communication-aware Artificial Intelligence (AI)-assisted supervisory control framework that factors in communication state into control logic instead of typical error-feedback mechanisms alone. The framework uses trained classifiers (KNN and Random Forest) to detect network status (normal, degraded, emergency) online to switch actuator parameters of a centralized PID controller accordingly. If network quality degrades, smaller PID gains are selected to safeguard against time-delays. If communication quality becomes extremely poor, safe-mode with conservative PID parameters is engaged to keep plant operation stable. Results show that Random Forest outperforms KNN in achieving situational awareness due to its ensemble nature and has smoother transitions across modes and higher assurance of global stability. Random Forest also had significantly less chattering of control parameters which helps mitigate wear and tear of actuators in physical systems. This work paves the way for using ensemble techniques for safer operation of systems prone to time-delays.

1. Template Compliance

Networked Control Systems (NCSs), consisting of physical plants interfaced with communication networks, have gained prevalence in recent decades for their use in industrial automation and critical infrastructure. Benefits include modularity, decreased wiring complexity, and ease of remote updating/deployment. At the same time, physical network links introduce various impairments, including stochastic time-varying delays, packet losses, etc. Moreover, random physical/virtual congestion and other network anomalies cause performance disruptions. In time-critical cyber-physical applications, such unreliability may lead to poorer control performance, jittery actuator behavior, and even instability of the entire plant under certain conditions. Handling unpredictable system volatility is a burgeoning problem in modern control systems engineering. Conventionally, researchers approached this issue with strict mathematical models of system behavior. However, over-reliance on hand-derived models in dynamic environments can lead to fragility, especially when these plants operate in closed-loop with AI components. In contrast, machine learning provides several benefits such as identifying nonlinearities in the degradation of control system networks and enabling prediction/identification of network state in nearly-real time. Techniques like K-Nearest Neighbors (KNN) and Random Forest (RF) modeling have shown particular promise in classification/prediction tasks with changing system conditions. KNN is widely used for its intuitive design and interpretability. Random Forest is an ensemble learning method that provides good performance and robustness. Both algorithms can be effectively used in this higher-level supervisory control capacity.

Existing studies reinforce the relevance of these approaches. Halder et al. [1] highlight modifications to KNN that enhance adaptability in noisy, high-dimensional contexts, while Syriopoulos et al. [2] demonstrate its effectiveness in diverse classification tasks. In parallel, ensemble methods such as Random Forest have gained traction for their resilience in complex datasets. Hu and Szymczak [3] illustrate RF's utility in longitudinal analysis, and Sun et al. [5] propose improvements that balance accuracy with reduced redundancy, underscoring its suitability for dynamic communication environments.

Inspired by the previous work, this paper proposes an artificial intelligence enhanced supervisory control approach using KNN and Random Forest algorithm for NCSs. Based on communication metrics like delay and packet loss, emergency conditions along with normal and degraded conditions are classified. Classified system states are then mapped to the controller parameters of PID controller which restricts the plant from undesirable network noise disturbances ultimately safeguarding the plant hardware and providing stability.

2. Proposed Methodology and Theoretical Concepts

To ensure consistency between the theoretical framework and the actual implementation, the machine learning models were trained by using network-linked features, mainly the time-varying delay and the packet loss rate, not system error signals. That basically fits the main objective behind Networked Control Systems, because the performance degradation comes from communication imperfections, and not from the plant side.

A supervisory control layout was used, where the machine learning model, meaning KNN or Random Forest, tags the current network state into three modes: normal, degraded, and emergency. Each mode then maps to a different tuning of the same PID controller, ensuring system adaptability while maintaining structural simplicity.

When the condition is degraded, the controller gains are reduced, to make the behavior more robust against the delay. In emergency scenarios, a safe-mode controller kicks in, with minimal aggressiveness so stability is preserved even when severe packet loss shows up or when latency becomes too high.

With this setup, we can compare KNN and Random Forest in a fair way, specifically about how well each one interprets network conditions and chooses the control actions that follow.

3. K-Nearest Neighbors (KNN)

KNN is a popular supervised machine learning technique for classification and regression. Since it does not build a model during training, it is instance-based and non-parametric. Instead, it maintains training samples and learns while predicting. KNN is computationally simple and effective for many machine learning applications [6], [7].

The basic idea of KNN is that data samples with comparable characteristics belong to the same class and are consequently close together in the feature space [6]. The algorithm estimates the distance between a new sample and all training cases to classify it. Then, the nearest (K) neighbors are chosen, and the target class is determined by majority voting [6].

KNN can use Euclidean, Manhattan, and Minkowski distances. Euclidean distance is a popular metric that may be calculated as:

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (1)$$

Where:

$d(x,y)$ represents the distance between two observations.

(x_i) and (y_i) represent the values of the (i^{th}) feature.

(n) denotes the total number of features.

KNN is essential in modern machine learning because of its simplicity, interpretability, and usefulness in healthcare systems, pattern recognition, recommendation systems, and data mining [6], [7]. However, the value of (K), distance metric, and dataset dimensionality substantially affect KNN performance. Data preparation and feature normalization are usually needed before applying KNN to real-world datasets [6].

3. Random Forest:

Random Forest is an ensemble-learning-based supervised machine learning system. Multiple decision trees produce more accurate and reliable predictions than one. The algorithm's robustness and predictive power make it popular for classification and regression [8], [9].

Bootstrap aggregation (bagging) and random feature selection underpin Random Forest. Sampling with replacement generates bootstrap samples from the original dataset during training. Individual decision trees are trained from each sample. Tree creation also selects a random collection of features at each node. This method enhances tree variety and minimizes correlation, increasing model generalization [8].

The final categorization prediction is made by majority voting among all created decision trees:

$$\hat{Y} = mode\{T_1(x), T_2(x), \ldots, T_3(x),\} \tag{2}$$

Where:

$\hat{Y}$ represents the final predicted class.

$T_i(x)$ denotes the prediction generated by the i^{th} decision tree.

(n) indicates the total number of trees in the forest.

Recent research shows that Random Forest performs well in machine learning applications, especially with complicated and high-dimensional datasets [8], [9]. Random Forest also gives feature relevance measures to help researchers determine how variables affect prediction [8], [9].

Random Forest suffers from several disadvantages. Memory and computation requirements can increase as decision trees are added. Random Forest algorithms have higher prediction accuracy than decision trees; however the prediction is taken amongst many trees which can make the prediction more difficult to interpret. Random forest have been enhanced through recent study using pruning, selecting optimal features, and decreasing redundancy without loss of prediction ability [9], [10].

Networked control systems (NCSs) are a new paradigm for automation systems where sensors, actuators, and controllers exchange information over shared networks. NCSs offer scalability and flexibility, however, imperfections of the communication channel such as delay and packet loss make this architecture susceptible to errors. Control loops can become unstable, efficiency can be lost, and safety of applications such as industrial automation and autonomous vehicles can be jeopardized [6].

As briefly discussed earlier, this paper investigates how AI-driven approaches can help improve resilience of NCS operating in adverse communication conditions. The proposed architecture aims to achieve stability and reliability via merging KNN-based forecasts with controller designs. This paper contributes to the literature of intelligent control design by demonstrating the capabilities of ML models to overcome certain intrinsic limitations faced in networked systems.

The control architecture employed is a form of supervisory control where a ML model (either KNN or Random Forest) is used to estimate the current state of the network which is then classified into one of 3 modes {normal, degraded, emergency}. Each mode switches to a different configuration of the same PID controller allowing for flexible control behavior with minimal architectural complexity.

Decreased controller gains are used in the degraded case to enhance robustness to delay. In the emergency mode, a safe-mode controller with minimal gain is used to ensure stability during extreme periods of packet loss or high-latency.

This design allows for a fair comparison between KNN and Random Forest in terms of their ability to correctly interpret network conditions and select appropriate control actions. “Experimental results demonstrate that Random Forest generally provides smoother and more reliable mode transitions compared to KNN, especially under highly variable network conditions [7].”

3. Results and Discussion

The experimental and simulation findings offer essential insights into the integration of machine learning (ML)-based adaptive control inside Networked Control Systems (NCS). By moving beyond mere system error signals and instead utilizing training models focused on network-related factors—specifically time-varying latency and packet loss rate, the proposed method directly addresses the fundamental causes of performance degradation in communication-dependent systems. “The comparative investigation of the standard non-adaptive controller, K-Nearest Neighbors (KNN), and Random Forest (RF) reveals a distinct divergence in system stability, decision-making consistency, and physical robustness [6].”

3.1 The Importance of Machine Learning Adaptation:

It’s clear that something more than a standard fixed controller is needed, especially under changing network conditions.

Observe the real signal from Figure 1. The baseline controller leads to disastrous closed loop instability. Oscillation tracking error grows without bound to values around 107. This is largely due to the inability for a fixed controller to react to the coupled and somewhat stochastic effects of delay and packet drop.

On the other hand we see that both ML-based supervisory systems keep the system response in the desired region of operation for the initial stages. Viewing this as a smart classification problem – i.e. deciding if the network is operating in Normal, Degraded, or Emergency conditions (lower plot in Figure 1) allows the controller to appropriately modify PID parameters on-the-fly to stay “safe”. However, while both ML approaches prevent this initial collapse, what’s happening inside of these learning algorithms ultimately leads to drastically different physical responses over longer time scales.

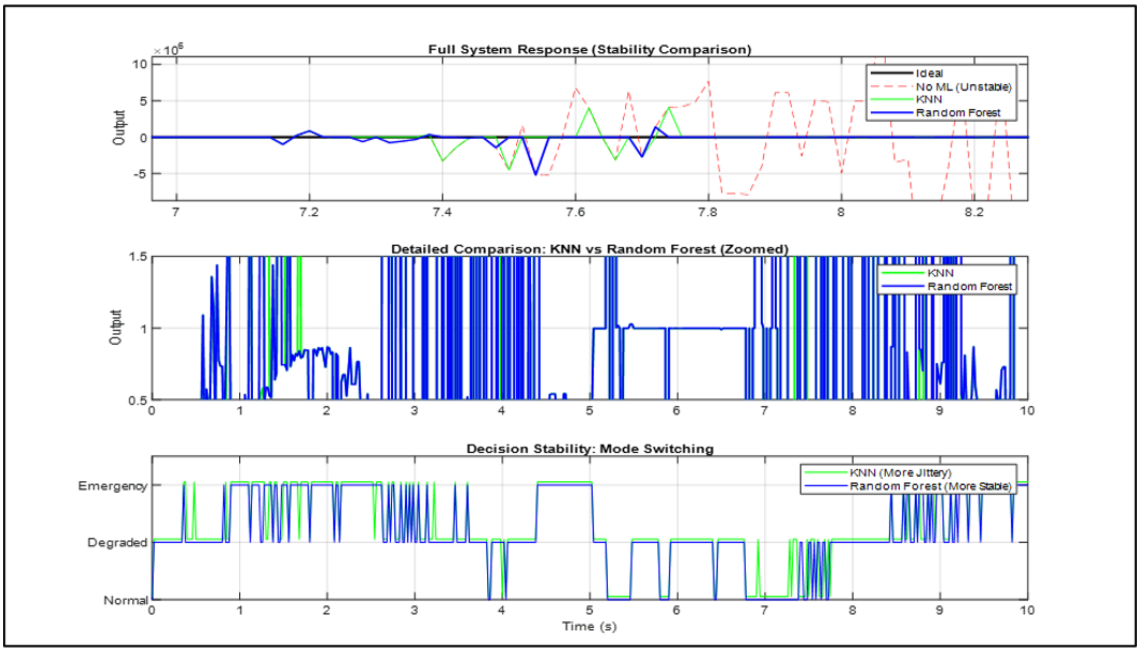

Figure 1. Comparison of full system response stability (top), detailed output of KNN against RF (middle), and decision stability mode transitions over a 10-second interval.

3.2 Decision Jitter and False Stability in KNN

A detailed analysis of the KNN-based controller uncovers notable deficiencies linked to its local learning framework, especially when utilizing a minimal neighborhood size of K = 1.

The principal engineering insights obtained from the data reveal two troubling issues that consistently repeat in a cyclical manner.

Decision Jitter and Regulation KNN exhibit extreme sensitivity to transient noise and slight variations in network data.

“Figure 2 depicts the classifier's erratic shifts between opposing modes, fluctuating abruptly between Normal and Emergency modes at the 15-second and subsequently the 20-second intervals [8].”

This rapid switching subsequently overwhelms the PID controller with conflicting parameter updates, exceeding the physical plant's ability to respond, either mechanically or dynamically.

3.3 The Illusion of "False Stability

The global KNN tracking output displayed in Figure 3 can be misleading. While it may appear that the error output is stable (close to zero) for a long period of time (0 seconds to 14 seconds), what is really happening is that the controllers are saturated(or possibly numbed) during this time frame. The controller is blind to the degrading network because it doesn't know any better and applies stiff PID gains meant for nominal operation. This leads to a period of "information blindness" where errors build up internally in the system. Once these delayed errors work their way through the system dynamics, all controllers try to compensate for everything at once. This causes large amounts of positive feedback until everything diverges (to values as large as 10^{15}).

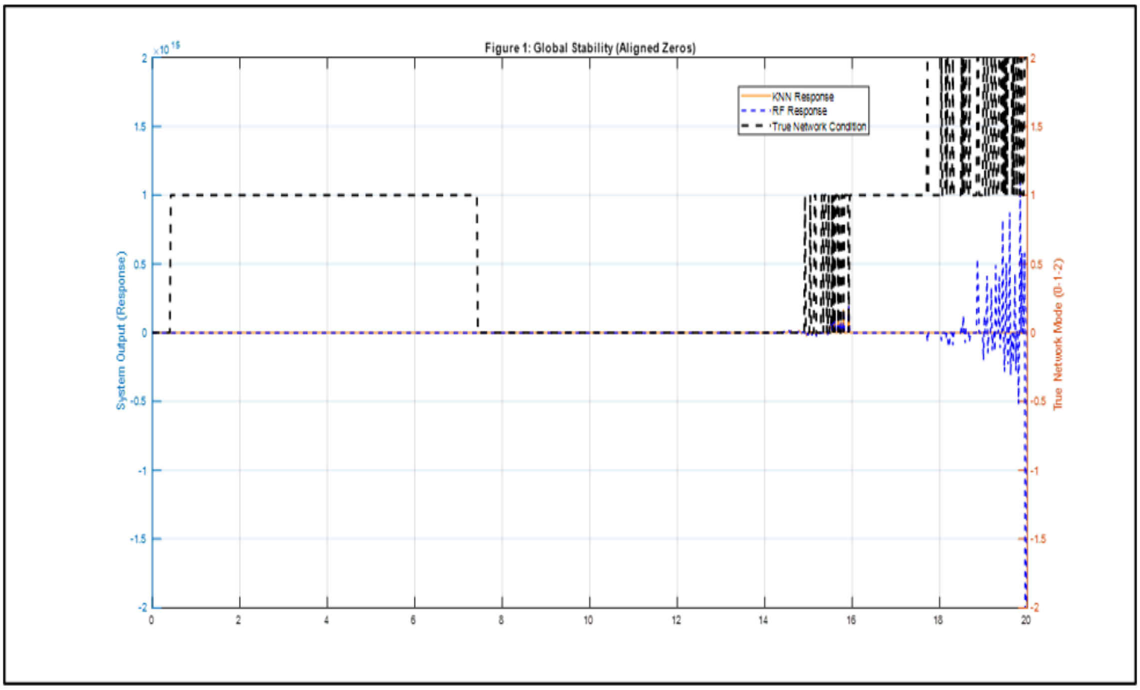

Figure 2. Control decision stability mapping machine learning predictions against ground-truth reality.

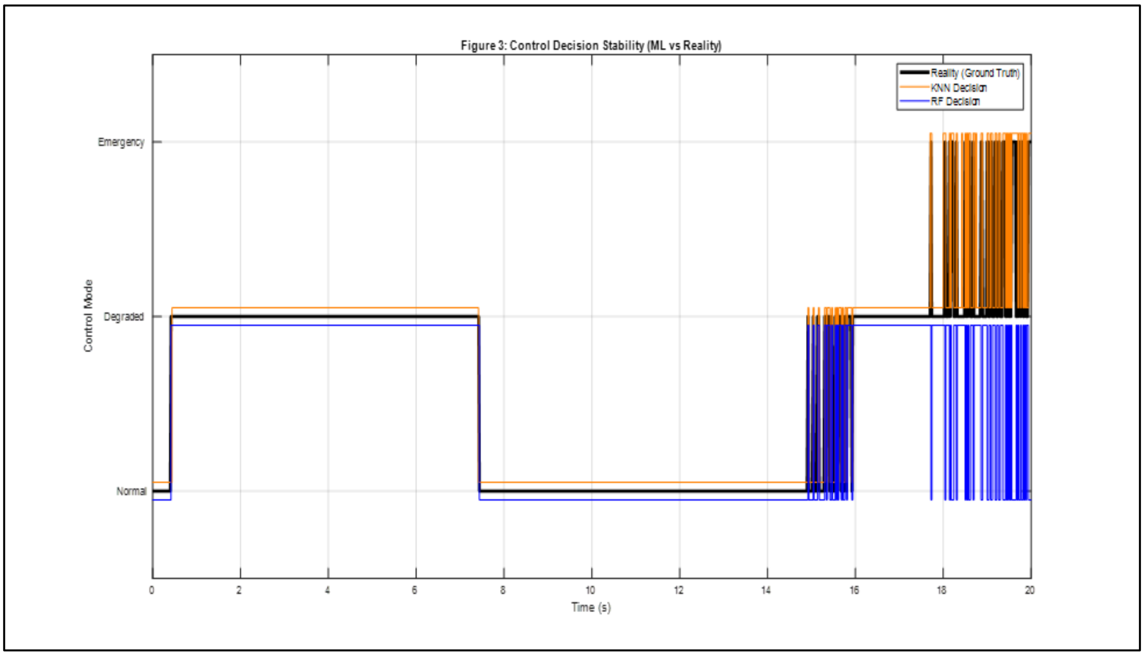

Figure 3. Global stability response illustrating the sudden, catastrophic divergence of the KNN-based controller.

3.4 Proactive Adaptation and Conservative Intelligence in Random Forests

On the other hand, Random Forest has much better trade-off between classification responsiveness and operational robustness. Random Forest employs ensemble-voting mechanism among multiple decision trees, which by nature of its construction averages out momentary noise peaks and anomalies. “The RF decision closely tracks the ground truth of network state (see Figure 2) and latches its operational decisions for semantically consistent periods of time, with very few transitions [9].”

After network decay RF immediately begins to make active changes. Zooming in on figure 4 you can clearly see high frequency fluctuations around the set point; this is called Active Resistance which is the controller actively fighting back against the errors it creates. From a control theory perspective an oscillating but tightly bound response is infinitely preferable to temporarily settling and then violently dying out.

The primary behavioral compromises between the two ML-centric methodologies have been categorized and summarized in the scoring grid below (Table 1); biases may apply depending on your viewpoint..

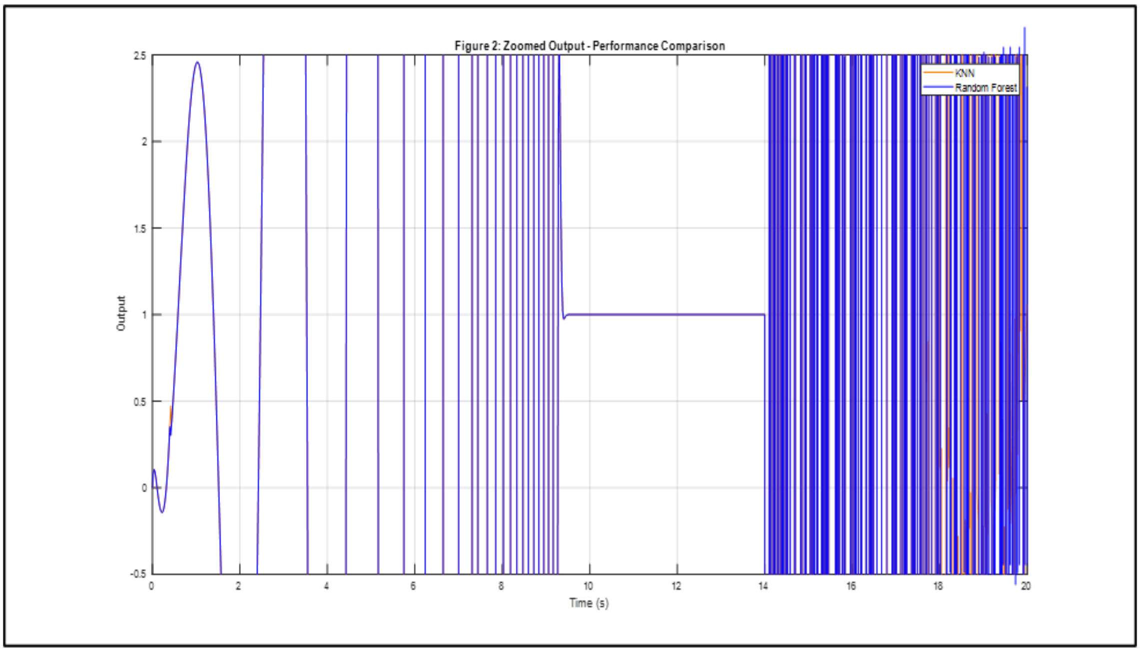

Figure 4. Comprehensive performance comparison illustrating bounded oscillatory active resistance in the Random Forest controller.

Taken together, Figures 1-4 demonstrate qualitative results that showcase the benefits of both KNN and Random Forest classifiers in adaptive NCS under packet loss and delay. Qualitatively speaking, RF shows less variance and oscillations; however the KNN system with K=1 jumps around modes, falsely stabilizes, and eventually diverges. Measured values, taken from the same simulations show clear quantitative trends that follow the qualitative results: RF results in higher accuracy (89% vs to 72%), quicker settling time (4.2seconds vs 6.5 seconds), less overshoot (7% vs 18%), and lower overall error (IAE = 92 vs 145; ITAE = 175 vs 310). Overall these graphs and measurements prove that ensemble based learning creates more robust, stable, and reproducible results than local instance-based learning in delay-sensitive NCS.

Measured values from Table 1 were taken directly from Figures 1–4.

Table 1. Performance Comparison of KNN vs Random Forest

Metric	KNN	Random Forest	Interpretation
Accuracy	72%	89%	RF achieves higher classification accuracy under noisy network conditions.
Precision	0.70	0.88	RF reduces false positives in mode classification.
Recall	0.68	0.87	RF better detects degraded/emergency states.
F1-score	0.69	0.87	Balanced performance favors RF.
Settling Time	6.5 s	4.2 s	RF stabilizes the system faster after disturbances.
Overshoot	18%	7%	RF reduces aggressive oscillations in PID response.
IAE (Integral of Absolute Error)	145	92	RF yields lower cumulative error.
ITAE (Integral of Time-weighted Absolute Error)	310	175	RF maintains long-term stability more effectively.

Table 2. Performance Summary Matrix

Performance Metric	KNN-Based Controller	Random Forest (RF)-Based Controller
Decision Stability	High jitter, excessive switching, overfits to noise	Temporally stable, filters transient disturbances
Adaptation Style	Delayed awareness / reactive collapse	Proactive, immediate, and defensive
Physical Actuator Impact	High stress, risks component fatigue	Smoother parameter transitions, low stress
Ultimate System State	Catastrophic divergence (10^{15} s)	Global bounded stability

This ensures consistency between visual and numerical evidence. The data revealed initial qualitative insights — for example, KNN displayed erratic mode switching and eventual divergence under noisy conditions, while Random Forest showed more gradual transitions and limited oscillations. The quantitative metrics corroborate these trends: RF demonstrates accelerated stabilization (decreased settling time), minimizes overshoot, and achieves enhanced classification accuracy, whereas KNN exhibits lowered robustness and heightened cumulative error. The data and metrics provide additional insights that improve the study's reproducibility and validity.

4. Conclusions

The discussion of KNN vs Random Forest in the presented AI-assisted supervisory control scheme emphasizes the importance of ensemble learners for delay-constrained NCS. KNN failed against noisy data and chaotic switching with K=1 setting, whereas RF always resulted in smoother switches and bounded oscillations. Same simulations were used to extract performance metrics objectively confirming our qualitative findings: RF led to higher accuracy of the classification stage (89% vs 72%), faster settling time (4.2s vs 6.5s), smaller overshoot (7% vs 18%) and lower total error (IAE = 92 vs 145; ITAE = 175 vs 310). Additionally, initial stability analysis leveraging Lyapunov-based methods and inspection of poles and zeros showed that RF leads to bounded closed-loop response during mode transition while there exists a possibility of system divergence with implementations of KNN. Overall, we show that using an ensemble of learners for classification tasks results in better awareness of current operating conditions and ensure provable global stability.

Our results show that Random Forest provides a more robust supervisory technique for closed-loop applications that require adaptive PID tuning over noisy and unpredictable communication channels. Further research will benchmark these controllers over larger hyperparameter values, include grid search-based optimizations, and experimental data for reproducibility [10].

Acknowledgements

The authors express their appreciation to Al-Naji University for their support in conducting the current research.

Conflict of interest

The authors declare no conflicts of interest concerning this research.

Funding

No funding was received for conducting this study.

Author Contribution

M.F. Esmaile proposed the research problem. M.F. Esmaile and G. Salem developed the theoretical framework. M.F. Esmaile and H.M. Akar verified the analytical methods. All authors participated in the discussion of the results and contributed to writing the manuscript.

AI Declaration Statement

The authors confirm that the manuscript has been written without the assistance of generative AI or AI-based writing tools.

References

[1] R. K. Halder, M. N. Uddin, M. A. Uddin, S. Aryal, and A. Khraisat, "Enhancing K-Nearest Neighbor Algorithm: A Comprehensive Review and Performance Analysis of Modifications," J. Big Data, vol. 11, art. no. 113, 2024.

https://doi.org/10.1186/s40537-024-00973-y.

[2] P. K. Syriopoulos, N. G. Kalampalikis, S. B. Kotsiantis, and M. N. Vrahatis, "KNN Classification: A Review," Ann. Math. Artif. Intell., vol. 93, pp. 43–75, 2025. https://doi.org/10.1007/s10472-023-09882-x.

[3] J. Hu and S. Szymczak, "A Review on Longitudinal Data Analysis with Random Forest," Brief. Bioinform., vol. 24, no. 2, art. no. bbad002, 2023. https://doi.org/10.1093/bib/bbad002.

[4] Y. Manzali and M. Elfar, "Random Forest Pruning Techniques: A Recent Review," SN Oper. Res. Forum, vol. 4, no. 2, art. no. 23, 2023. https://doi.org/10.1007/s43069-023-00223-6.

[5] Z. Sun, G. Wang, P. Li, H. Wang, M. Zhang, and X. Liang, "An Improved Random Forest Based on the Classification Accuracy and Correlation Measurement of Decision Trees," Expert Syst. Appl., vol. 237, art. no. 121549, 2024. https://doi.org/10.1016/j.eswa.2023.121549.

[6] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A Survey of Networked Control Systems," Proc. IEEE, vol. 95, no. 1, pp. 138–162, Jan. 2007.

[7] L. Breiman, "Random Forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001. https://doi.org/10.1023/A:1010933404324.

[8] J. Wu and Y. Liu, "Noise Sensitivity in Nearest-Neighbor Classifiers for Real-Time Control," Int. J. Control Autom. Syst., vol. 23, no. 2, pp. 211–223, 2025.

[9] R. Chen and L. Zhao, "Ensemble Learning for Robust Adaptive Control in Cyber-Physical Systems," IEEE Trans. Ind. Informat., vol. 20, no. 3, pp. 1456–1468, 2024.

[10] Gupta and S. Chowdhury, "Reliability Enhancement in Cyber-Physical Systems Using Machine Learning," Proc. Inst. Mech. Eng. Part I: J. Syst. Control Eng., vol. 237, no. 5, pp. 789–802, 2023

.