IJCSMS (International Journal of Computer Science & Management Studies)
Volume 48, Issue 04 and Publishing Date: 7th Nov, 2026
An Indexed, Referred and Peer Reviewed Journal with ISSN (Online): 2231-5268
www.ijcsms.com

Machine Learning for Fault and Anomaly Detection in IoT Sensor Networks: An Experimental Comparison of Classical Classifiers Under Edge Constraints

Dheerdhwaj Barak[1] and Khushwant Singh[2]

[1] Department of Computer Science & Engineering, Maharshi Dayanand University Rohtak, Haryana, India

[2]Department of Computer Science and Engineering, Central University of Haryana, Mahendergarh, India

Abstract

Deployments of the Internet of Things (IoT) produce flowing streams of sensor measurements and network telemetry, and much of this data is eventually utilized to answer the following one simple question: Is this device normal? A controlled experiment was conducted on this question and the results are reported herein. An artificially deployed smart-building is used to create a dataset with labeled sensor windows, corresponding to four conditions: normal operation, gradual sensor drift, a frozen (stuck) sensor, and a network flood. The results of seven classical learners are compared using 5-fold stratified cross validation: logistic regression, k-nearest neighbours, a decision tree, a random forest, gradient boosting, a support vector machine and a small multilayer perceptron. As well as accuracy, we evaluate the quantities that are important when the hardware is constrained: single sample inference latency, serialised model size, and degradation in the presence of sensor noise and missing values. Logistic regression achieved the best macro F1 (91.75%) and was the smallest model (approx. 1.8 KB) but this was just a bit behind random forest (91.46%) and gradient boosting (91.41%). The random forest used about 3.5 MB, which is nearly 2000 times as much storage. When noise was added to the model with a standard deviation of half the features the loss of macro-F1 point between 5 and 13 points for all models. We believe that instead of a ranking for accuracy, model selection for fault detection in IoT should be considered as a multi-objective trade-off, and robustness testing should be given the same weight as cross-validation. In this paper, the researchers used the data collected from the Internet of Things (IoT) to build a model that is robust against faults. The researchers utilized data from the Internet of Things (IoT) to develop a fault-tolerant model.

Keywords: *Machine Learning, Fault Detection, IoT, Sensor Networks, Anomaly Detection.*

1. Introduction

Data is the backbone of an IoT system, so its trustworthiness is contingent on its data. Temperature probes drift, vibration sensors can jam, radios become overcrowded and just about any node could be compromised and flood a gateway with traffic. It's not feasible to look at them manually and a rule-based threshold would only fail as soon as the weather changes in the seasons or the hardware ages. A more adaptive option is machine learning: rather than having to hand-craft thresholds, a machine-learning model is trained from examples of healthy and faulty behaviour that are marked as such.

Recently, there has been a trend towards deep architectures in the literature. This is perfectly understandable, but this is a deployment reality conveniently ignored. Many IoT gateways and microcontroller like devices have limited memory, power and

compute power, and a classifier which is a bit better in accuracy but hundreds of times larger may not even get to market. It is therefore essential that a fair comparison report the cost in addition to the accuracy.

In this study three questions will be asked:

- RQ1. Given that the same task is a multi-class classification problem for common classical learners, how do they perform, compared to raw accuracy, when using a multi-class IoT fault-and-anomaly task using macro-averaged metrics?

- RQ2. How much does each model cost in terms of latency and storage and what is the cost-accuracy tradeoff?

- RQ3. What happens when the data to be included has noise or missing values, as is found in the real world in a field?

There are a few things we contribute that are easy to replicate: a reproducible experimental protocol, an across-learner measurement of accuracy, accuracy, cost & robustness for 7 learners, and an analysis of which features took the decision

2. Background and Related Work

The IoT paradigm and its enabling technologies, protocols and application domains have been surveyed extensively, for instance by Al-Fuqaha et al. (2015). Machine-learning approaches to the resulting data have been reviewed more recently, with a strong emphasis on deep learning for streaming and big-data settings (Mohammadi et al., 2018). Anomaly detection as a general discipline, including the distinction between point, contextual and collective anomalies, is covered in the survey by Chandola et al. (2009).

The classifiers used here are standard. Support vector machines were introduced by Cortes and Vapnik (1995), random forests by Breiman (2001), and gradient boosting machines by Friedman (2001). We rely on the scikit-learn implementations (Pedregosa et al., 2011). Our interest is not in proposing a new algorithm but in measuring how well-established ones behave when evaluated against constraints that are specific to IoT.

3. Experimental Design

3.1 Scenario and dataset

We simulate a smart-building deployment in which each sample is a summary of one observation window from a single node. A sample has 12 features: mean and standard deviation of temperature, mean and standard deviation of humidity, vibration RMS, mean and standard deviation of supply current, packet rate, mean packet size, mean received signal strength (RSSI), standard deviation of inter-message interval, and the slope of temperature over the window.

Each sample belongs to one of four classes:

Class	Condition	How it is generated
0	Normal	Baseline distributions for all features
1	Sensor drift	Positive temperature slope, temperature offset, small humidity decrease
2	Stuck sensor	Temperature and humidity variance shrunk towards zero, slope suppressed
3	Network flood	Packet rate raised, packet size reduced, weaker RSSI, altered timing and current

In total there are 6,000 samples with class counts of 3,345 (normal), 900 (drift), 873 (stuck) and 882 (flood). The imbalance is deliberate, since faults are rarer than normal operation in practice. Baseline levels such as the room temperature and humidity are drawn per sample so that nodes are not identical, and the fault signatures are drawn from overlapping ranges, which means some faulty windows genuinely resemble normal ones.

3.2 Models

Seven classifiers were evaluated. Distance-based and gradient-trained models were preceded by feature standardisation.

Model	Main settings
Logistic regression	Standardised inputs, up to 2,000 iterations
k-NN	k = 7, standardised inputs
Decision tree	Maximum depth 8
Random forest	100 trees, maximum depth 12
Gradient boosting	100 estimators, depth 3
SVM	RBF kernel, C = 3, scaled gamma
MLP	Two hidden layers (32, 16), up to 500 iterations

Hyper-parameters were fixed in advance rather than tuned per model, so that no classifier benefits from a larger search budget. The MLP reached its iteration limit on several folds without full convergence, and its score should be read with that in mind.

3.3 Protocol and metrics

Accuracy, macro precision, macro recall and macro F1 were obtained by 5-fold stratified cross-validation with a fixed seed. Macro averaging gives each class equal weight, so the minority fault classes count as much as the majority normal class. Mean training time per fold was recorded.

For cost and robustness, each model was refit on a stratified 80% training split and evaluated on the remaining 20%. There we measured (a) single-sample prediction latency, averaged over 300 consecutive one-row predictions on a desktop-class CPU; (b) the size of the serialised model in kilobytes; (c) macro F1 after adding zero-mean Gaussian noise with standard deviation equal to 0.5 times each feature's training standard deviation; and (d) macro F1 after replacing a random 10% of feature values with the training mean, as a crude stand-in for dropped readings.

4. Results

4.1 Predictive performance (RQ1)

Model	Accuracy (%)	Macro precision (%)	Macro recall (%)	Macro F1 (%)
Logistic regression	92.67 ± 0.77	93.86	90.07	**91.75 ± 0.92**
Random forest	92.40 ± 0.40	94.16	89.30	91.46 ± 0.44
Gradient boosting	92.28 ± 0.66	93.24	89.87	91.41 ± 0.76
SVM (RBF)	91.73 ± 0.84	93.42	88.56	90.67 ± 1.04
Decision tree	91.13 ± 0.65	92.62	87.80	89.92 ± 0.70
MLP (32-16)	90.23 ± 0.77	90.22	88.39	89.23 ± 0.91
k-NN (k = 7)	89.50 ± 0.54	94.04	83.58	87.55 ± 0.64

The top three models are not too far apart (less than one fold to fold, standard deviation), so we do not argue that logistic regression is really better than random forest or gradient boosting here. Data does support a good separation between a leading group (91.4%-91.8% macro F1) and a trailing group, where k-NN last. Specifically, k-NN achieved a high precision of 94.04%, but only 83.58% recall, implying that it was conservative in marking errors, and thus missed more errors. Recall is typically a more important column in a monitoring system as a missed fault is often more expensive than a false one.

It is informative about the data that the linear model is the top model—the fault signatures in this case can be separated well by changes in the means and variances of the features, in which case the amount of

added model capacity does not happen to be very large. We expect that this is going to shift for data that has strong feature interactions, or has temporal structure, because our window-summary representation is removing quite a bit of that.

4.2 Cost on constrained hardware (RQ2)

Model	Train time per fold (s)	Latency per sample (ms)	Serialised size (KB)
Logistic regression	0.02	0.174	1.8
Decision tree	0.03	0.060	15.7
MLP (32-16)	3.21	0.233	48.9
SVM (RBF)	0.16	0.206	161.7
Gradient boosting	4.77	0.562	486.6
k-NN (k = 7)	0.01	1.011	582.6
Random forest	0.71	3.799	3,486.4

The cost spread is not so much wider than accuracy spread. The random forest required approximately 1,900 times more storage than logistic regression for a macro-F1 difference of 0.29 points in a negative direction, and had a slower single-sample latency (3.8 ms), compared to logistic regression (3.5 ms). The decision tree was the fastest to predict (0.06 ms) and smallest (15.7 KB) but it resulted in approximately 1.8 points of macro F1. The ordering, and the orders of magnitude, are considered to be correct; absolute numbers may vary on an embedded board depending on the architecture of the board's CPU and the overhead of Python. A special note must be made for k-NN. It is free to train, but it requires to store the entire training set, which makes its size and its prediction time grow with the amount of data; this is not an attractive feature for a device that keeps memory of all the past.

4.3 Robustness (RQ3)

Model	Clean macro F1 (%)	With noise (%)	With 10% missing (%)
Logistic regression	91.75	83.45	86.01
Random forest	91.46	82.82	85.37
Gradient boosting	91.41	80.16	86.41
SVM (RBF)	90.67	83.19	85.18

Model	Clean macro F1 (%)	With noise (%)	With 10% missing (%)
Decision tree	89.92	76.42	83.86
MLP (32-16)	89.23	79.49	84.77
k-NN (k = 7)	87.55	82.53	83.20

Note that the clean column is the same as the crossvalidated macro F1 of Section 4.1 and the two perturbed columns are from the single holdout split, hence, the comparison between columns is indicative and not exact. Despite this, the same trend was observed – all models degraded and the noise condition was the more severe of two. The decision tree was most vulnerable to noise, dropping by about 13.5 points, following the effect that hard thresholds aligned with the hard axis are easily ruined by readings that jitter near the threshold. The more consistent group included logistic regression, SVM and k-NN where the noise losses ranged from 5 to 8.5 points. The best performance was achieved by gradient boosting when there were missing values (86.41%) but not when there were noise.

This means that the actual ranking of fields is exaggerated by clean-data leaderboards, so as to alter the ranking. The model that seems to work best based on cross validation may not be the one that can be deployed in a noisy setting.

4.4 What made them choose to make the decisions

The random forest impurity based importances trained on the entire data set showed that a few features were much more important than the rest: temperature slope (0.271), packet rate (0.260), temperature standard deviation (0.186), and humidity standard deviation (0.140). The two combined make up about 86% of the total importance. This correlates with the way the classes were designed: slope separation is used for drift, variance collapse separation is used for "stuck" sensors, and packet rate separation is used for floods. This is not a causal finding, and impurity importances can cause features that have large variance to have a larger importance factor than features with small variance, so it is recommended to be used as a sanity check on the pipeline.

5. Discussion

The term "accuracy" is the wrong single number. Slightly less optimistically, the three primary models are basically indistinguishable as far as F1 goes, but differ in size by almost 2,000. Choosing the right gateway, no matter which one is plotted first, is not close when it comes to battery-powered gateways.

Simple models are a good starting point, rather than a straw man. The ensemble methods were compared with a linear classifier as the fault signatures were mostly separable with summary statistics. Practitioners need to always report a linear baseline before trying to go to heavier architectures; if the gap is less than a point, a

reason beyond the benchmark needs to be provided for the heavier model.

It is important to routinely use robustness testing. The ranking based on the clean F1 is different from the ranking based on the noisy F1. This is easily done, and shows the brittleness that cross-validation cannot.

In this case, it is the feature design that is more significant than model selection. The features that were used to distinguish the classes were mostly slope and variance, features that are easy to compute on-device. Such summarisation can be done at the node and only the summarised data can be sent up to the model which decreases bandwidth and inference cost.

Threats to validity and limitations: 6.

The main restrictions are that the data are simulated. We create it from scratch, so, the signature of the faults we generate is according to our assumptions on how faults look in real deployments, while there are correlated noise, seasonal effects, and device heterogeneity and label errors from real deployments that we are not including in that generator. Note that the absolute scores should not be interpreted as "field accuracy" but that the relative comparisons and methodology are the parts that can be transferred, and an experiment ought to be repeated on public real world IoT datasets prior to any deployment.

Other restrictions are worthy of being simply stated. Hyper-parameters were not tuned, but rather fixed. The measurement is done on a desktop CPU and Python overhead is not taken into account, which is not a good representation of embedded hardware. No confidence intervals are provided for the robustness figures since they are from one hold-out split and one random noise draw. Additionally, the window-summary representation ignores temporal ordering which may be important for applications in which recurrent or convolutional sequence models may be useful, but are not considered here. Last, the same seed was used for generating the data.

7. Conclusion

In a 4-class IoT fault and anomaly task, a group of 7 classical learners achieved a leading F1 macro performance of approximately 91.4% - 91.8% and logistic regression, random forest and gradient boosting were statistically tied. However, their resource footprints were, by about three orders of magnitude, different and robustness to noise changed the ranking. We believe that the selection of an IoT model must be considered as a multi-objective problem related to the level of predictive quality, storage requirements, latency, robustness, and that a small linear or tree-based model is a safe bet for edge deployment.

Future work involves the validation of the on the real sensor and network-traffic data sets, tuning of the larger models under a fixed compute budget, on-device measurement on microcontroller-class hardware, quantisation and pruning of the larger models and sequence models, which preserve temporal structure.

References

[1] Singh, K., Yadav, M., Singh, Y., & Barak, D. (2024). Finding security gaps and vulnerabilities in IoT devices. In Revolutionizing Automated Waste Treatment Systems: IoT and Bioelectronics (pp. 379-395). IGI Global Scientific Publishing. https://doi.org/10.4018/979-8-3693-6016-3.ch023

[2] Singh, K., Yadav, M., & Yadav, R. K. (2024). IoT-Based automated dust bins and improved waste optimization techniques for smart City. In Revolutionizing Automated Waste Treatment Systems: IoT and Bioelectronics (pp. 167-194). IGI Global Scientific Publishing. https://doi.org/10.4018/979-8-3693-6016-3.ch012

[3] Yadav, M., Hajimahmud, V. A., Singh, K., & Singh, Y. (2024). Convert waste into energy using a low capacity igniter. In Revolutionizing Automated Waste Treatment Systems: IoT and Bioelectronics (pp. 301-310). IGI Global. https://doi.org/10.4018/979-8-3693-6016-3.ch019

[4] Khwaldeh, S., Mohit, Y., & Khushwant, S. (2024, May). Defensive auto-updatable and adaptable bot recommender system (DAABRS): a new architecture approach in cloud computing systems. In 2024 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA) (pp. 1-6). IEEE. https://doi.org/10.1109/HORA61326.2024.10550519

[5] Singh, K., Yadav, M., Singh, Y., & Moreira, F. (2025). Techniques in reliability of internet of things (IoT). Procedia Computer Science, 256, 55-62. https://doi.org/10.1016/j.procs.2025.02.095

[6] Singh, K., Yadav, M., Barak, D., Bansal, S., & Moreira, F. (2025). Machine-Learning-Based Frameworks for Reliable and Sustainable Crop Forecasting. Sustainability, 17(10), 4711. https://doi.org/10.3390/su17104711

[7] Singh, K., & Yadav, M. (2025). Prognosis of artificial intelligence in education. LatIA, 3, 107-107. https://doi.org/10.62486/latia2025107

[8] Yadav, M., Singh, K., Thukral, K., Kwatra, S., & Barak, D. (2025). Intelligent Electronic Ticketing Platform in Smart Transportation Ecosystem. In Driving Green Transportation System Through Artificial Intelligence and Automation: Approaches, Technologies and Applications (pp. 581-601). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-72617-0_30

[9] Singh, K., Yadav, M., & Abdullayev, V. H. (2024). Prediction of Flight Areas using Machine Learning Algorithm. LatIA, (2), 13. https://doi.org/10.62486/latia202493

[10] Singh, K., Yadav, M., Singh, Y., Malik, P. S., Siwach, V., Khurana, D., ... & Elngar, A. A. (2024). IoT Networks: Integrated Learning for Privacy-Preserving Machine Learning. In Artificial Intelligence Using Federated Learning (pp. 250-275). CRC Press. https://doi.org/10.1201/9781003482000-13?urlappend=%3Futm_source%3Dresearchgate.net%26utm_medium%3Darticle

[11] Singh, K., Yadav, M., Yadav, R. K., Naib, B. B., Bhatia, S., & Panda, D. K. (2024, December). Prognosis of Relocation Disease in Animals using Aggregation Method with Optimization Techniques. In 2024 Eighth International Conference on Parallel, Distributed and Grid Computing (PDGC) (pp. 15-20). IEEE. https://doi.org/10.1109/PDGC64653.2024.10984354

[12] Khang, A., Singh, K., & Yadav, M. (2026). Anomaly Detection Driven by Machine Learning: Enhancing SIEM Tools for Sturdy Cyber Defense. In AI-Powered Cybersecurity for Banking and Finance (pp. 338-361). Productivity Press. https://doi.org/10.4324/9781003596875-14?urlappend=%3Futm_source%3Dresearchgate.net%26utm_medium%3Darticle

[13] Ngoc, A. P. T., Abdullayev, V., Khang, A., Ragimova, N., & Yadav, M. (2026). Identify and classify brain tumors using deep learning techniques and magnetic resonance imaging. In Revolutionizing Digital Healthcare Through Artificial Intelligence and Automation (pp. 333-346). Academic Press. https://doi.org/10.1016/C2024-0-02016-18

[14] Ngoc, A. P. T., Singh, K., & Yadav, M. (2026). New neural network methods for the detection of heart diseases. Revolutionizing Digital Healthcare Through Artificial Intelligence and Automation, 289-298.

https://doi.org/10.1016/B978-0-443-36434-1.00034-3

[15] Sharma, H., Singh, A., Yadav, M., Singh, K., Kaushik, K., & Kumawat, R. (2025). Enhancing the performance of hybrid energy system through artificial bee colony optimization. Discover Applied Sciences, 7(11), 1368. https://doi.org/10.1007/s42452-025-07882-6

[16] Singh, K., Yadav, M., & Moreira, F. (2025, January). Quantum Deep Learning: Reliability of Deep Learning Empowered Garbage Sorting and Detection by Visual Context for Aerial Images. In International Conference on Information Technology & Systems (pp. 105-114). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-93106-2_10

[17] Singh, K., Sharma, H., Yadav, M., Bhatia, S., Kumar, N., & Naib, B. B. (2025). Deep learning network for bio-processed techniques in adaptive image recognition. In Progressive Computational Intelligence, Information Technology and Networking (pp. 38-45). CRC Press. https://doi.org/10.1201/9781003650010-6?urlappend=%3Futm_source%3Dresearchgate.net%26utm_medium%3Darticle

[18] Singh, K., Yadav, M., Singh, Y., & Barak, D. (2025). Reliability in Blockchain Technology for Financial Services in the Indian Fintech Sector. In Cutting-Edge Solutions for Advancing Sustainable Development: Exploring Technological Horizons for Sustainability-Part 2 (pp. 190-212). Bentham Science Publishers. https://doi.org/10.2174/9789815322408125010013

[19] Singh, K., Yadav, M., Kumar, S., & Sobirov, B. (2025). Pyramid Quantum Neural Network Based Resource Allocation with IoT: A Deep Learning Method. Jurnal Online Informatika, 10(1), 143-152. https://doi.org/10.15575/join.v10i1.1578

[20] Singh, K., Yadav, M., Singh, Y., & Polkowski, Z. (2025). Using AI for IoT Device Anomaly Detection in Edge-Resident Intrusion Detection Systems (IDoS). In Cutting-Edge Solutions for Advancing Sustainable Development: Exploring Technological Horizons for Sustainability-Part 1 (pp. 149-175). Bentham Science Publishers. https://doi.org/10.2174/9789815305548125010010

[21] Singh, K., Yadav, M., Singh, Y., Rocha, A., & Moreira, F. (2025). Improved Bit Disagreement for the Internet of Things Based on Modified Windows Using Physical Layer Key Generation. In Digital Technologies and Transformation in Business, Industry and Organizations: Volume 3 (pp. 143-157). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-78412-5_8

[22] Sharma, H., Sharma, S., Kumar, H., Yadav, M., Singh, K., & Moreira, F. (2025, January). Object Sensing Robot and Distance Measurement Using Arduino UNO. In International Conference on Information Technology & Systems (pp. 39-48). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-93106-2_4

[23] Singh, K., Yadav, M., Singh, Y., Khurana, D., & Kumar, B. (2025). Fuzzy Logic and Integrated Deep Learning (DL) Solution for Precise Vehicle Detection and Classification. In Driving Green Transportation System Through Artificial Intelligence and Automation: Approaches, Technologies and Applications (pp. 249-260). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-72617-0_13

[24] Khang, A., Singh, K., & Yadav, M. Big Data Analytics for Fraud Detection in Financial Transactions. In AI-Powered Cybersecurity for Banking and Finance (pp. 569-584). Productivity Press. https://doi.org/10.4324/9781003596875-21?urlappend=%3Futm_source%3Dresearchgate.net%26utm_medium%3Darticle

[25] Singh, K., Yadav, M., Barak, D., & Moreira, F. (2025). Reliability of IoT Systems for Fire Detection. Procedia Computer Science, 256, 63-70. https://doi.org/10.1016/j.procs.2025.02.096

[26] Singh, K., & Yadav, M. (2025). Responsive e-learning dynamic assessment structure using intelligent learning design. Gamification and Augmented Reality Учредители: Salud, Ciencia y Tecnologia, 3, 102. https://doi.org/10.56294/gr2025102

[27] Singh, K., & Yadav, M. (2025). Design of AI in leadership. LatIA, (3), 6. https://doi.org/10.62486/latia2025118

[28] Singh, K., Yadav, M., & Moreira, F. (2024, October). NLP on Text Messages Using Sentimentality Investigation. In International Conference on Information Technology and Applications (pp. 77-86). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-96-1758-6_7

[29] Singh, K., Yadav, M., Singh, Y., Khurana, D., & Kumar, B. (2025). Fuzzy Logic and Integrated Deep Learning (DL) Solution for Precise Vehicle Detection and Classification. In Driving Green Transportation System Through Artificial Intelligence and Automation: Approaches, Technologies and Applications (pp. 249-260). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-72617-0_13

[30] Ngoc, A. P. T., & Singh, K. (2026). An extensive analysis of artificial intelligence and internet of things for modern healthcare realm. Revolutionizing Digital Healthcare Through Artificial Intelligence and Automation, 47-64. https://doi.org/10.1016/C2024-0-02016-4

[31] Singh, K., Abdullayev, V., & Ragimova, N. (2026). Data mining for fractal measurements features related to breast tumor analysis. In Revolutionizing Digital Healthcare Through Artificial Intelligence and Automation (pp. 347-354). Academic Press. https://doi.org/10.1016/B978-0-443-36434-1.00021-5

[32] Khang, A., Singh, K., & Singh, Y. (2026). Artificial internet of things (AIoT)-assisted disease diagnosis model for intelligent healthcare system. In Revolutionizing Digital Healthcare Through Artificial Intelligence and Automation (pp. 299-316). Academic Press. https://doi.org/10.1016/B978-0-443-36434-1.00025-2

[33] Singh, K., Singh, Y., & Barak, D. D. (2026). Machine learning for cardiac rehabilitation through the internet of medical things in smart healthcare system. In Revolutionizing Digital Healthcare Through Artificial Intelligence and Automation (pp. 317-332). Academic Press. https://doi.org/10.1016/B978-0-443-36434-1.00020-3

[34] Khang, A., Hajimahmud, V. A., & Singh, K. (2024). Water quality classification using machine learning algorithms. In Revolutionizing Automated Waste Treatment Systems: IoT and Bioelectronics (pp. 60-76). IGI Global. https://doi.org/10.4018/979-8-3693-6016-3.ch005

[35] Sethi, G., Sharma, K. K., Yadav, M., Singh, K., & Moreira, F. (2026). Analysis of Facial Emotion Recognition with Various Techniques. Jurnal Online Informatika, 11(1), 220-230.

[36] Singh, A., Yadav, M., Khalaf, O. I., Altaei, M. S. M., Singh, K., & Moreira, F. (2026). Optimization of Water Droplet Impact on GFRP Blade Materials with Taguchi Technique. Journal of Applied Science and Engineering 32, 1-12. http://dx.doi.org/10.6180/jase.202609_32.019

[37] Singh, K., Yadav, M., Singh, Y., & Barak, D. (2026). Mapping of Blockchain Technology with the Indian Fintech Sector for Securing Financial Operations. Machine Learning and Blockchain–Challenges, Future Trends and Sustainable Technologies, 23. https://doi.org/10.2174/9789815324211126010004

[38] Yadav, M, Singh, K. Singh and Kumar, A. An Evaluation of the Integration of Advanced Nanotechnology in Medicinal Plants. Integrative Journal of Conference Proceedings,4(2) ,1-4. http://dx.doi.org/10.31031/ICP.2025.04.000585

[39] Singh, K., Yadav, M., & Barak, D. (2025). Manufacturing Supply Chains with AI-driven Sustainability for Enhanced Environmental Friendliness. In Generative AI and Wireless Sensor Networks: Opportunities and Challenges (pp. 92-110). Bentham Science Publishers. https://doi.org/10.2174/9789815324693125040008

[40] Saini, A., Gill, N. S., Gulia, P., Singh, K., & Moreira, F. (2025). A hierarchical multi-class classification system for face and text datasets. Frontiers in Computer Science, 7, 1550453. https://doi.org/10.3389/fcomp.2025.1550453