JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)

VOL. 11. NO. 4 MAY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i4.7481

EVALUATION OF ANN- LEVENBERG MARQUARDT MODELS FOR FAULT DETECTION IN SMART FARMING SYSTEM

Luh Kesuma Wardhani[1,3]; Agus Buono[1]; Sri Wahjuni[1*]; Muhamad Syukur[2]

Doctoral Study Program of Computer Science, School of Data Science, Mathematics and Informatics[1]
Department of Agronomy and Horticulture, Faculty of Agriculture[2]
IPB University, Bogor, Indonesia[1,2]
https://www.ipb.ac.id[1,2]
luhkesuma@apps.ipb.ac.id, agusbuono@apps.ipb.ac.id, my_juni04@apps.ipb.ac.id*,
muhsyukur@apps.ipb.ac.id

Study Program of Informatics, Faculty of Science and Technology[3]
UIN Syarif Hidayatullah, Jakarta, Indonesia[3]
https://www.uinjkt.ac.id[3]
luhkesuma@uinjkt.ac.id

(*) Corresponding Author
(Responsible for the Quality of Paper Content)

Abstract— *Sensor readings in open field monitoring systems are influenced by disruptions, degradation, and operational unreliability. These conditions may result in inaccurate data and unreliable system decisions. However, existing studies focus on detection accuracy and rarely examine the trade-off between detection performance and computational efficiency of Artificial Neural Networks trained using the Levenberg–Marquardt algorithm (ANN–LM) in smart farming environments. This study evaluates the fault-detection capability of ANN–LM for soil moisture sensor readings by analyzing both detection performance (accuracy, precision, recall, and F1-score) and computational efficiency (execution time, CPU usage, and memory consumption), thereby addressing the trade-off between performance and efficiency. Baseline data, hypothetical dataset that represent the soil moisture reading from a smart chilli pepper farming system in normal operating conditions, were used to generate fault-injected datasets representing four common faults: drift, bias, spike, and malfunction. The ANN–LM model was evaluated under five fault-detection scenarios with different network architectures. Model performance was evaluated using accuracy, precision, recall, and F1-score, while computational cost was assessed through execution time, CPU usage, and memory usage. The results show that ANN–LM achieves an accuracy of 0.996–0.999, precision of 1.000, recall of 0.987–1.000, and F1-scores of 0.992–1.000 across all scenarios. Simple ANN architectures give accuracy of 0.997 with reduced execution time (33.74 seconds) and lower CPU usage (50.50%) compared to more complex architectures that require 591.88 seconds and 78.40% CPU usage. Therefore, these results indicate point out that ANN–LM is suitable for smart agricultural systems under resource-constrained conditions.*

Keywords: *Artificial Neural Network, Fault Detection, Internet of Things (IoT), Levenberg–Marquardt, Smart Farming.*

Intisari— *Pembacaan sensor pada sistem pemantauan di lahan terbuka sering dipengaruhi oleh berbagai gangguan, degradasi perangkat, dan ketidakandalan operasional. Kondisi ini dapat menyebabkan data menjadi tidak akurat dan berdampak pada pengambilan keputusan sistem yang kurang tepat. Sebagian besar fokus penelitian saat ini terletak pada tingkat akurasi deteksi dan belum banyak mengkaji trade-off antara kinerja deteksi dan efisiensi komputasi khususnya pada Artificial Neural Network yang dilatih menggunakan algoritma Levenberg–Marquardt (ANN–LM) dalam konteks pertanian cerdas. Penelitian ini mengevaluasi kemampuan deteksi gangguan dari ANN–LM pada data sensor kelembaban tanah dengan menganalisis dua* *faktor utama, yaitu kinerja deteksi (accuracy, precision, recall, dan F1-score) serta efisiensi komputasi (waktu eksekusi, penggunaan CPU, dan konsumsi memori), sehingga dapat memberikan gambaran mengenai trade-off antara performa dan efisiensi. Data baseline berasal dari pembacaan kelembaban tanah pada kondisi normal dari sistem irigasi otomatis pada budidaya cabai di lahan terbuka. Data ini kemudian digunakan untuk menghasilkan dataset melalui prosedur fault injection. Fault buatan ini mewakili empat jenis fault yang umum muncul di pertanian: drift, bias, spike, dan hardware malfunction. Model dievaluasi dalam lima skenario dengan arsitektur jaringan yang berbeda. Hasil menunjukkan akurasi 0,996–0,999, presisi 1,000, recall 0,987–1,000, dan F1-score 0,992–1,000. Arsitektur yang lebih sederhana mencapai akurasi 0,997 dengan waktu eksekusi lebih singkat (33,74 detik) dan penggunaan CPU 50,50%, lebih efisien dibandingkan arsitektur kompleks. Temuan ini menunjukkan bahwa ANN–LM sesuai untuk sistem pertanian cerdas dengan sumber daya terbatas.*

***Kata Kunci**: Jaringan Saraf Tiruan, deteksi kesalahan, Internet of Things (IoT), Levenberg–Marquardt, Pertanian Cerdas.*

INTRODUCTION

Digital technology is transforming modern agriculture. Modern agriculture uses digital technology, The Internet of Things (IoT) and artificial intelligence (AI) to enhance agricultural sustainability and productivity [1], [2], [3]. IoT and AI provide real-time data collection from sensors and smart devices. Sensors that measure soil moisture, temperature, air humidity, and other environmental conditions are key for providing critical information for system decisions. However, sensor data is not always reliable. Faulty data can be resulted from disruptions, which could interfere with automated decisions in precision agriculture [4]. Therefore, fault detection mechanisms are very important to ensure the reliability of smart agricultural systems. Existing studies have highlighted the importance of reliable sensor data for smart farming and agricultural IoT systems. Several fault detection approaches have been proposed to increase the reliability of sensor reading as an important part of decision making. The authors in [5] have developed a context aware fault diagnostic scheme which able to identify low intensity faults, such as drift, bias, spikes, and data loss, in wireless sensor networks.

This study demonstrates that the approaches can enhance the reliability of sensor measurements. Similar soft faults used in this study are often observed in agricultural IoT sensors. It indicates the need for fault-diagnosis mechanisms that prevent corrupted sensor data being utilized in agricultural decision-making processes. In another study, the authors in [6] reviewed the current status and prospects of sensor fault diagnosis in the agricultural Internet of Things, as well as diagnostic methods based on statistical analysis, machine learning, and signal processing. They conclude that it remains challenging for achieving accurate, real-time, and resource-efficient diagnosis in harsh and dynamic agricultural environments, which motivates the development of the fault-detection approach presented in this study. Smart farming architectures critically depend on continuous sensor measurements from IoT networks to enable real-time monitoring and control. Research of [7] analyzed IoT-based smart irrigation systems using Fault Tree Analysis, revealing that sensors exposed to harsh environments are highly susceptible to failures, which can propagate to system-wide unsafe states without explicit diagnosis mechanisms.

This vulnerability underscores the need for robust fault-handling methods, such as the fuzzy entropy-based approach proposed here In addition, the authors in [8] provided a critical analysis of smart farming which is based on IoT and AI, covering sensors, communication technologies, cloud platforms, and machine-learning. This research shows that AI model performance in agriculture depend on the reliability of sensor data, but, on the other hand, the persistent issues with data acquisition and quality remain unresolved. These findings show the importance of reliable fault detection mechanisms in smart farming. ANNs have been applied widely in smart farming environments because of their ability to model complex and nonlinear relationships in sensor data. ANN models can identify complex fault patterns that using conventional methods. Furthermore, the integration of deep learning algorithms with IoT systems has been shown to improve sensor reliability and support more adaptive and intelligent smart farming solutions [9] such as optimization of IoT node placement which increase energy effeciency [10].

Even though ANN models have good performance, they still have limitation with optimization strategies. Improper optimization can result in poor convergence and unstable learning behavior, specially when dealing with large amount

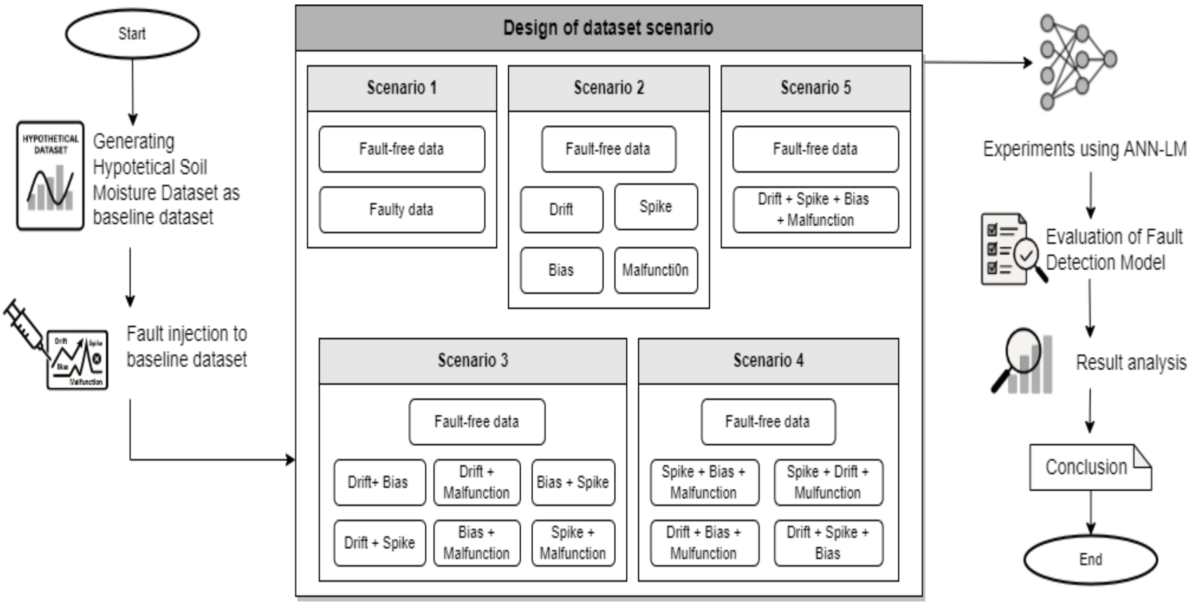
Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 11. NO. 4 MAY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i4.7481

nonlinear sensor data [11]. To overcome this limitation, more effective optimization methods are required. The Levenberg–Marquardt (LM) algorithm has been introduced as an enhanced training algorithm for ANN. LM is known for its fast convergence and stability in handling large nonlinear datasets. ANN-LM models achieved high-level accuracies with strong correlations between actual and predicted values [12]. This research evaluated the performance of ANN-LM algorithm for sensor fault detection in smart farming environment. Several experiments are conducted to assess accuracy and computational cost performance. This study is not intended to compare ANN–LM with other fault-detection algorithms, it focuses to develop a more reliable sensor fault detection system and evaluating the ANN's performance efficiency trade off to address the challenges of data inaccuracies in smart farming system.

MATERIALS AND METHODS

This section discuss about the material and methods used in this study. We will discuss about the experiment design, data preparation, ANN–LM configuration, and evaluation methodology used in this study. The experiment design includes generating hypothetical soil moisture sensor data, fault injection procedures, dataset layout scenario design, ANN-LM model training, and performance evaluation on accuracy and computational cost. This approach follows common practices in data intensive modelling for agriculture and smart industry [13], [14]. The overall research workflow adopted in this study is illustrated in Figure 1.



Source : (Research Results, 2026)

Figure 1. Research Flow Diagram

The dataset used in this study is a hypothetical data which represent the soil moisture sensor reading in a smart chilli pepper farming system in normal operating conditions. Ten thousand data points were used as baseline data, which was generated with reference to a normal soil moisture range of 60–80% [15]. To represent fault data, fault injection procedure was carried out using the research baseline data. The four types of faults used were drift, bias, spike, and malfunction, as defined in [5]. The hypothetical dataset with fault injection provides space for more controlled experiments, especially when real data is difficult to obtain or does not cover the entire range of interference variations. This strategy is often used in AI research for critical systems because it helps to build models that are more resilient to data uncertainty [6].

The purpose of using a hypothetical dataset was to control fault injection and systematically assess result of various dataset scenarios. Although the fault categories are defined individually, the multi-class and mixed-fault scenarios implicitly represent fault conditions, where multiple faults characteristics exist and interact within the same dataset. The controlled fault injection enables an accurate study of model performance under specific fault situations that are challenging to gain in real-world. Nevertheless, this approach does not fully capture the environmental complexity which present in actual agricultural fields, and real sensor data may exhibit more complex noise characteristics.

In the ANN-LM model, the basic architecture is a multilayer feedforward ANN with one to two hidden layers. To determine the best architecture for fault detection in each scenario, a grid search was used. This procedure determined the number of neurons in the hidden layers to gain the best performance. The dataset is split into training, validation, and testing datasets. Sixty percent of the data is used for training, 20% for validation, and 20% for testing. Using this composition, the model can be evaluated objectively and will not be biased to any specific dataset. This technique improved the model's generalization and performance, therefore the evaluation can be conducted correctly [16]. It is important to evaluate the robustness of the model from a complete dataset, including synthetic fault events [17].

RESULT AND DISCUSSION

Result

Table 1 and Table 2 summarize the experiment result for performance metrics (accuracy, precision, recall, and F1-score) and computational costs (execution time, CPU usage, and memory consumption). As illustrated in Figure 1. The ANN–LM model was evaluated under five scenarios to assess fault detection performance Scenario 1 represents a binary classification experiment, in which the output classes were fault-free and faulty data. Scenarios two to five represent multi-class classification that distinguish between fault-free and mixed faulty data.

In scenario 1, a binary classification was performed to distinguish between fault-free and faulty data. Three ANN models were evaluated: model 1A with architecture [1 12], model 1B with architecture [1 12 4] and model 1C with architecture [1 12 8]. All three models achieve an accuracy of 0.997, a precision of 0.995, and a recall of 0.992. These results show that the model can distinguish between fault free and faulty data well, even though there are still a small number of faults that are not correctly detected.

Table 1. Result of Performance Evaluation

Scena-rio		Archi-tecture	Accu-racy	Precision	Recall	F1-Score
1	A	[1 12]	0.997	0.995	0.992	0.994
	B	[1 12 4]	0.997	0.995	0.992	0.994
	C	[1 12 8]	0.997	0.995	0.992	0.994
2	A	[1 3 8]	0.999	1.000	0.997	0.999
	B	[1 3 12]	0.999	1.000	0.997	0.999
	C	[1 3 4]	0.998	1.000	0.992	0.996
3	A	[1 5 0]	0.997	0.997	0.987	0.992
	B	[1 5 4]	0.997	0.997	0.987	0.992
	C	[1 5 8]	0.997	0.997	0.987	0.992
4	A	[1 5]	0.997	0.987	0.997	0.992
	B	[1 5 4]	0.997	0.987	0.997	0.992
	C	[1 5 8]	0.997	0.987	0.997	0.992
5	A	[1 3 4]	0.997	1.000	1.000	1.000
	B	[1 3 8]	0.997	1.000	1.000	1.000
	C	[1 3]	0.996	0.999	1.000	0.999

Source : (Research Results, 2026)

Scenario 2 demonstrates improved performance with the following architectures: model 2A [1 3 8], model 2B [1 3 12] and model 2C [1 3 4]. Both model 2A and model 2B achieved an accuracy of 0.999, a precision of 1.000, and a recall of 0.997, whereas model 2C obtained an accuracy of 0.998, a precision of 1.000, and a recall of 0.992. These results show that the architectures in scenario 2 are effective in minimizing false positive classifications, and these architectures resulted in reliable fault detection.

Scenario 3, which includes architectures [1 5 0], [1 5 4], and [1 5 8], achieved consistent results with an accuracy of 0.997, precision of 0.997, and recall of 0.987. The recall was lower than in other scenarios, which indicates that certain faults were not successfully detected. It showed that the limitation is mostly driven by data characteristics rather than network size. Scenario 3 consists of more complex fault patterns which reduce class separability and introduce ambiguity in the feature space. As a result, the ANN–LM model tends to maintain high precision by avoiding false positives, but at the cost of missing some actual faults. Different performance patterns are observed in Scenario 4. Precision result decreases to 0.987, recall and accuracy remain at 0.997. This outcome shows that the model in Scenario 4 successfully detects faults and produces more positive prediction errors. This behavior arises from the impact of the fault data characteristics in Scenario 4, where the fault patterns are quite different from the normal pattern but still show partial overlap among fault classes. Therefore, the ANN–LM model tends to detect as many faults as possible, which explains the high recall but also leads to an increase in false positive predictions.

Scenario 5 shows the best performance, whereas models 5A [1 3 4] and 5B [1 3 8] achieved perfect results on all metrics (accuracy, precision, and recall = 1.0). Model 5C [1 3] was also almost perfect with an accuracy of 0.996, precision of 0.999, and recall of 1.0. These results confirm that the architecture in scenario 5 achieved the best balance for fault detection in the scope of this study. detection.

VOL. 11. NO. 4 MAY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i4.7481

Overall, the performance results show that the ANN–LM model achieves consistently high accuracy across all scenarios; on the other hand, recall and precision vary with fault complexity. Recall is affected by presence of several faults and sometimes become uncertain, conversely, single fault achieves high and nearly perfect detection. Utilizing the correct dataset scenario is important to achieve a more reliable fault-detection system.

Another parameter that needs to be calculated from the fault-detection task is the computational cost, namely execution time, CPU usage, and memory usage. Table 2 is summarize experiment result of computational cost evaluation. In scenario 1, models 1A [1 12], 1B [1 12 4], and 1C [1 12 8] need long execution time of 591.88 seconds, CPU usage of 78.40%, and memory usage of 4.80%. From these result, it is shown that scenario 1 needs high computational cost, because large number of neurons in the hidden layers require increased iterations of weight updates to achieve convergence. Hence this process results low computational efficiency.

Scenario 2 shows better results, with an execution time of 58.84 seconds, CPU usage of 86.00%, and memory usage of 4.81%. The increasing CPU usage shows the model is effective for resource optimization, faster computations, and shorter execution time. The execution time of scenario 3 is lower than that of scenario 1, which was 185.21 seconds, with CPU usage of 81.60% and memory usage of 4.81%. It shows balanced results between execution time and resource utilization, though they are less efficient than those in scenario 2. From computational costs perspective, the higher cost in scenario 3 is due to increased fault pattern complexity. Therefore, the network requires more iteration during classification.

Table 2. Result of Computational Cost

Scenario		Archi-tecture	Execution Time	CPU Usage (%)	Memory Usage (%)
1	A	[1 12]	591.88	78.40	4.80
	B	[1 12 4]	591.88	78.40	4.80
	C	[1 12 8]	591.88	78.40	4.80
2	A	[1 3 8]	58.84	86.00	4.81
	B	[1 3 12]	58.84	86.00	4.81
	C	[1 3 4]	58.84	86.00	4.81
3	A	[1 5 0]	185.21	81.60	4.81
	B	[1 5 4]	185.21	81.60	4.81
	C	[1 5 8]	185.21	81.60	4.81
4	A	[1 5]	157.28	82.50	4.81
	B	[1 5 4]	157.28	82.50	4.81
	C	[1 5 8]	157.28	82.50	4.81
5	A	[1 3 4]	33.74	50.50	4.81
	B	[1 3 8]	33.74	50.50	4.81
	C	[1 3]	33.74	50.50	4.81

Source : (Research Results, 2026)

According to Table 2, the computational cost of scenario 4 is higher than scenario 3 with CPU usage reached 82.50%, execution time was 157.28 seconds, and memory usage was 4.81%. These results indicate a trade off between fault complexity and network configuration. The increased efficiency may be due to its fault patterns that are easier to distinguish than those in Scenario 3. As a result, the ANN–LM model achieves faster computation while handling complex fault conditions. The best computational cost among all evaluated scenarios had resulted from scenario 5, with an execution time of 33.74 seconds, the lowest CPU usage at 50.50%, and the lowest memory usage at 4.81%. Scenario 5 has simple network architecture and fault patterns, then ANN-LM model can achieve faster convergence and hence reduce computational costs.

Scenario 5 works best when implemented using edge devices because it achieves high accuracy and low computational cost. Scenario 2 is also very promising, as it reduces execution time by utilizing more CPU resources. In contrast, scenario 1 is the least efficient, with the longest execution time and the resource use. Execution time, CPU usage, and memory usage reported in this study are interpreted as indirect indicators of latency, energy demand, and hardware feasibility. The shorter execution time relates to lower latency, and lower CPU usage indicates a lower processing cost. Furthermore, the model shows its capability to operate reliably on resource constrained platforms.

These results indicate that the scalability of the ANN–LM model is determined by architectural design rather than the learning algorithm. As the network size increases, execution time and CPU usage grow larger. However, simple ANN architectures with fewer neurons can achieve high detection accuracy and reduce computational cost. From scalability perspective, ANN–LM can be adjusted to edge-based smart farming systems by regulating model complexity. It should be noted that the high accuracy reported in this study is obtained under controlled experimental conditions, where faults are injected according to a model from previous work [saeed]. Therefore, the results should be interpreted as an evaluation of model behaviour under well-defined fault scenarios rather than as a direct representation of real-world field performance.

Discussion

This subchapter discusses the results of experiments on the performance of ANN-LM in detecting faults in soil moisture sensors. According to the experiments, the best performance was obtained in models 5A–5C in scenario 5, a dataset of four fault combinations. These results were superior to models 3A–3C in scenario 3, which involved only two faults. This finding can be explained by the fact that data with four faults produced a more contrasting distribution than baseline data, so that the model are easy to classify the classes. On the other hand, combining two faults often causes ambiguous patterns that reduce the recall value. It is in line with [18] which confirms that clear and contrasting variations can enhance classification performance. One interesting aspect to discuss is the variation in recall values between scenarios.

Models 3A–3C in scenario 3 result low recall because some faulty data were not successfully detected. This problem may reduce the reliability of the precision agriculture system. Conversely, models 4A–4C in scenario 4 have achieved high recall, so almost all faults can be identified, even though precision is slightly reduced. This condition is align with other study that in the context of fault detection, false positives are more tolerable than false negatives [19]. Inability to detect faults may increase risk and directly affect operational decisions, such as in irrigation settings. Therefore, it is important to maintain a high recall in fault detection system design. Models 5A–5C in scenario 5 achieve an optimal combination of high recall and maintained precision, making them the most realistic solution for real-world implementation.

The experiment shows a correlation among network size, computational efficiency, and overall performance. Models 2A–2C with a 3–12 neuron architecture achieved high accuracy, reduced execution time, and lower CPU usage. For example, models 2A and 2B achieved an accuracy of 0.999 with an execution time of 58.84 seconds. In contrast, models 1A–1C in scenario 1 required 591.88 seconds, even though they did not achieve higher accuracy. The pattern is consistent with previous findings [16], indicating that the LM algorithm can enhance training, even on a simple network.

Computational efficiency is important in smart agricultural system, especially when deploying fault detection systems which involves limited memory and processing power edge devices. In this context, [17] emphasizes the importance of efficient models for maintaining system stability on limited resources hardware. A 60-20-20 data split was also adopted to increase the robustness of the experimental results, with larger validation and test datasets employed to avoid biasing performance evaluation across fault scenarios with small sample sizes [17], [20].

In general, the results of this study have wide implications for smart farming applications. First, the results show that models 2A–2C and 5A–5C achieve high performance with a simple structure, therefore they can be deployed on edge nodes. Second, the computational cost observed in the ANN–LM model supports its deployment in real-time multiple-fault detection, where response time and resource constraints become essential concerns. Third, the results show that the ANN–LM model has stable performance even when trained with a limited amount of data.

The main contribution of this study is to provide empirical proof that ANN–LM can achieve reliable performance with modest computational cost when detecting a multi-fault scenario with simple network architectures. These findings highlight the ANN–LM as a practical multi-fault-detection method, which can be implemented in a smart agricultural system because of its efficiency, robustness, and the ability to operate on limited infrastructure. Furthermore, it is suggested that ANN–LM can be feasibly deployed on edge devices for smart agricultural applications, provided that appropriate network configurations are selected to balance detection performance and computational constraints.

CONCLUSION

The results show that ANN–LM achieves good results on performance evaluations across all scenarios. Increasing network complexity does not yield better detection results, as simpler architectures often achieve performance comparable to more complex ones. Variations in detection metrics across scenarios are mostly caused by fault characteristics rather than by network size. From a computational cost point of view, we found that execution time and CPU usage are affected by network architecture. A larger network architecture has higher computational cost without proportional performance gains. In contrast, compact ANN architectures, particularly in Scenarios 2 and 5, provide a balance between detection accuracy and computational efficiency, shorter execution time, and consistent memory usage. In general, these findings confirm that ANN–LM can provide reliable multiple fault-detection performance and that computation remains efficient when properly configured. This study validates that ANN–LM is appropriate for a smart agricultural system under resource-constrained conditions. This study has the potential to be developed in multiple areas for future research. First, to obtain more realistic real-world IoT failure behaviors and model robustness, more complex

VOL. 11. NO. 4 MAY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i4.7481

factors can be considered, such as environmental noise, sensor heterogeneity, and various field conditions commonly encountered in smart agricultural deployments for multiple fault combinations. Second, to get a clear picture of fault detection in a smart agricultural system, the model will be tested using real soil moisture sensor data and run on an edge device. This experiment is needed to obtain an overview of system stability under limited energy and memory resources. Finally, comparative benchmarking against alternative ANN training algorithms and non-ANN fault-detection methods will be conducted to provide a broader performance context.

REFERENCE

[1] C. Arya, "Iot Based Precision Farming and Agriculture - Aspects and Technologies," *Mathematical Statistician and Engineering Applications*, vol. 70, no. 2, pp. 1426–1433, 2021, doi: 10.17762/msea.v70i2.2335.

[2] N. Jadhav, B. Rajnivas, V. Subapriya, S. Sivaramakrishnan, and S. Premalatha, "Enhancing Crop Growth Efficiency Through IoT-enabled Smart Farming System," *Eai Endorsed Transactions on Internet of Things*, vol. 10, 2023, doi: 10.4108/eetiot.4604.

[3] N. Vemuri, N. Thaneeru, and V. M. Tatikonda, "Smart Farming Revolution: Harnessing IoT for Enhanced Agricultural Yield and Sustainability," *Journal of Knowledge Learning and Science Technology Issn 2959-6386 (Online)*, vol. 2, no. 2, pp. 143–148, 2023, doi: 10.60087/jklst.vol2.n2.p148.

[4] Y. S. Tan, L. W. Chew, Y. X. Tan, and S. Z. Tan, "IoT-based Smart Farming System," *International Journal of Emerging Multidisciplinaries Computer Science & Artificial Intelligence*, vol. 3, no. 1, 2024, doi: 10.54938/ijemdcsai.2024.03.1.270.

[5] U. Saeed, Y.-D. Lee, S. U. Jan, and I. Koo, "CAFD: Context-Aware Fault Diagnostic Scheme towards Sensor Faults Utilizing Machine Learning," *Sensors*, vol. 21, no. 2. 2021, doi: 10.3390/s21020617.

[6] X. Zou *et al.*, "Current Status and Prospects of Research on Sensor Fault Diagnosis of Agricultural Internet of Things," *Sensors*, vol. 23, no. 5, pp. 1–25, 2023, doi: 10.3390/s23052528.

[7] M. M. Rahman, A. Abdulhamid, S. Kabir, and P. Gope, "Failure Analysis of IoT-based Smart Agriculture System: Towards Sustainable Food Security," in *2023 5th International Conference on Sustainable Technologies for Industry 5.0 (STI)*, 2023, pp. 1–6, doi: 10.1109/STI59863.2023.10464772.

[8] S. Qazi, B. A. Khawaja, and Q. U. Farooq, "IoT-Equipped and AI-Enabled Next Generation Smart Agriculture: A Critical Review, Current Challenges and Future Trends," *IEEE Access*, vol. 10, pp. 21219–21235, 2022, doi: 10.1109/access.2022.3152544.

[9] X. Yuan, Y. Fan, C. Zhou, X. Wang, and G. Zhang, "Research on Twin Extreme Learning Fault Diagnosis Method Based on Multi-Scale Weighted Permutation Entropy," *Entropy*, vol. 24, no. 9, p. 1181, 2022, doi: 10.3390/e24091181.

[10] J. S. Prashanth, G. B. Krishna, A. V. K. Prasad, and P. R. Rao, "Smart Farming Revolution: A Cutting-Edge Review of Deep Learning and IoT Innovations in Agriculture," *SN Operations Research Forum*, vol. 6, no. 1, pp. 1–39, 2025, doi: 10.1007/s43069-025-00434-z.

[11] T. Nguyen, H.-B. Ly, H. Thi, and V. Q. Tran, "On the Training Algorithms for Artificial Neural Network in Predicting the Shear Strength of Deep Beams," *Complexity*, vol. 2021, no. 1, 2021, doi: 10.1155/2021/5548988.

[12] S. Rao *et al.*, "Day-Ahead Load Demand Forecasting in Urban Community Cluster Microgrids Using Machine Learning Methods," *Energies*, 2022, doi: 10.3390/en15176124.

[13] P. Agarwal, J. I. M. Gonzalez, A. Elkamel, and H. Budman, "Hierarchical Deep LSTM for Fault Detection and Diagnosis for a Chemical Process," *Processes*, vol. 10, no. 12, p. 2557, Dec. 2022, doi: 10.3390/pr10122557.

[14] H. O. Choe and M. Lee, "Artificial Intelligence-Based Fault Diagnosis and Prediction for Smart Farm Information and Communication Technology Equipment," *Agriculture*, vol. 13, no. 11, p. 2124, 2023, doi: 10.3390/agriculture13112124.

[15] S. Wahjuni, W. Wulandari, and M. Kholili, "Development of Fuzzy-Based Smart Drip Irrigation System for Chili Cultivation," *JUITA: Jurnal Informatika*, vol. 10, no. 1, p. 115, May 2022, doi: 10.30595/juita.v10i1.12998.

[16] K. A. Kharusi, A. E. Haffar, and M. Mesbah, "Adaptive Machine-Learning-Based Transmission Line Fault Detection and Classification Connected to Inverter-Based Generators," *Energies*, vol. 16, no. 15, p. 5775, 2023, doi: 10.3390/en16155775.

[17] V. Borja-Jaimes, M. Adam-Medina, B. López-Zapata, L. G. V Valdés, L. C. Pachecano, and E. M. Sánchez-Coronado, "Sliding Mode Observer-Based Fault Detection and Isolation Approach for a Wind Turbine Benchmark," *Processes*, vol. 10, no. 1, p. 54, 2021, doi: 10.3390/pr10010054.

[18] G.-J. Qi and J. Luo, "Small Data Challenges in Big Data Era: A Survey of Recent Progress on Unsupervised and Semi-Supervised Methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 4, pp. 2168–2187, 2022, doi: 10.1109/tpami.2020.3031898.

[19] G. Wrat, P. Ranjan, S. K. Mishra, J. T. Jose, and J. C. Das, "Neural Network-Enhanced Internal Leakage Analysis for Efficient Fault Detection in Heavy Machinery Hydraulic Actuator Cylinders," *Proceedings of the Institution of Mechanical Engineers Part C Journal of Mechanical Engineering Science*, vol. 239, no. 3, pp. 1021–1031, 2024, doi: 10.1177/09544062241289309.

[20] M. K. AlOmar, F. Khaleel, A. A. AlSaadi, M. M. Hameed, M. A. AlSaadi, and N. Al-Ansari, "The Influence of Data Length on the Performance of Artificial Intelligence Models in Predicting Air Pollution," *Advances in Meteorology*, vol. 2022, pp. 1–20, Sep. 2022, doi: 10.1155/2022/5346647.

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri