International Journal of Computer Trends and Technology
ISSN: 2231–2803 / https://doi.org/10.14445/22312803/IJCTT-V74I4P102

Volume 74 Issue 4, 10-28, April 2026

Original Article

Decentralized Agentic AI Orchestration for Autonomous Self-Healing and Resilience in Distributed Cyber-Physical Systems

Abdinasir Ismael Hashi[1], Abdirizak Mohamed Hashi[2], Osman Abdullahi Jama[3], Ibrahim Rashid Abdullahi[4]

[1,3,4]*Computer Science, Somali National University, Mogadishu, Somalia.*
[2]*Computer Science, Jazeera University, Mogadishu, Somalia.*

[1]*Corresponding Author : nasirhaji@snu.edu.so*

Received: 25 February 2026 Revised: 30 March 2026 Accepted: 13 April 2026 Published: 27 April 2026

Abstract *- The study explores a decentralized agentic artificial intelligence orchestration paradigm for enhancing self-recovery and resilience within a distributed CPS. The traditional paradigms, characterized by their centralized nature, have been observed to present constraints like low scalability, delays in reaction, and susceptibility to single-point failures. As a result, the study has proven to be unsuitable for application in dynamic and complex settings. To solve these problems, a decentralized agentic AI framework is proposed, which utilizes autonomous agents with deep learning models, such as CNN, RNN, and the combined CNN-RNN model for anomaly detection and recovery. According to experimental findings, while the CNN and RNN models exhibit perfect recall performance (1.0000), the study further exhibits high false positives and low reliability scores. Conversely, the hybrid model performs far better than the individual models with regard to F1-score (0.9863), almost perfect AUC score, reduced false positive rates (0.0476), and reliability levels. Despite its relatively high detection latency compared to the individual models, the hybrid framework offers the best trade-off between precision, reliability, and stability.*

Keywords *- Decentralized Agentic AI, Cyber-Physical Systems, Anomaly Detection, Hybrid CNN-RNN, System Resilience, Self-Healing Systems.*

1. Introduction

Intelligent technology's rapid development has fundamentally changed the design of modern systems as well as how they are used and maintained. Over the last few years, "Cyber-Physical Systems (CPS)", which combine computation and networking with the physical world, have become foundational elements in many areas, including but not limited to intelligent transportation systems, smart cities, healthcare, industrial automation, etc. CPS operate in an environment where digital components continuously interact with the physical world [1]. Because of the dynamic nature of CPS, it is inherently more complex and continues to increase in both scale and heterogeneity. As CPS continue to grow, there is an increasing study challenge in addressing how to ensure CPS are reliable, resilient, and able to recover from failures [2].

However, traditional centralized control models are no longer adequate to handle distributed and complex infrastructures. In addition, centralized models are often associated with single points of failure, scalability limitations, and slower response times, particularly for real-time and mission-critical systems [3]. In recent times, there has been a significant increase in interest in decentralized systems where decision-making capabilities are distributed across multiple autonomous components of the system. In this regard, the introduction of Agentic Artificial Intelligence (Agentic AI) has introduced a major leap forward [4]. Agentic AI refers to intelligent systems that are comprised of autonomous agents that are capable of perceiving their environment, making decisions, learning from experiences, and taking action independently. Unlike traditional AI models that operate independently of one another, Agentic AI models are designed to collaborate and coordinate with one another in real-time. In addition, each of the agents acts independently as a decision-making entity while at the same time contributing to the overall system goals [5]. Such a model is particularly well-suited for distributed CPS systems where a number of components must work together in coordination without a central authority.

The concept of orchestration is of critical importance in facilitating collaboration among these autonomous agents. The concept of decentralized agentic AI orchestration is a dynamic coordination of these agents to ensure that they are able to share information and respond to different conditions in a collaborative manner [6]. In addition, the orchestration mechanism is adaptive and cognizant of the environment to ensure that the operations of the system are optimal despite the dynamic conditions of the environment. One of the most important capabilities that is enabled by the use of decentralized agentic AI is the concept of autonomous self-healing [7]. In a complex environment such as a CPS, it is possible for different types of failures to occur, such as hardware failure, network failure, cyberattacks, or even environmental factors. Self-healing is a critical concept that has been proposed for these environments to ensure that the operations of the system are not adversely affected by these types of failures. Through dynamic monitoring and decision-making, it is possible for the agentic AI to identify the causes of the failure and make the appropriate corrections to ensure that the operations of the system are optimal [8].

Another closely related concept to self-healing is resilience (the ability of a system to endure, adjust to, and recover from disruptions). Resilience does more than recover from failures; it maintains acceptable performance levels even under unfavorable circumstances [9]. The distribution of intelligence across the system increases resilience when using decentralized agentic AI, thus preventing total system failure if one or more of the individual components fail. Furthermore, the learning capabilities programmed into agents would enable the entire system to adapt its response strategies as time goes on. Therefore, it would have increased capability against future disruptions [10]. However, implementing decentralized agentic AI orchestration within a CPS also has some significant issues. These include establishing a secure means of communicating between agents, guaranteeing uniformity in decision-making across multiple locations, managing the added burden of computation, and addressing the ethical implications of making decisions autonomously. In addition, there continues to be a need for study to develop scalable architectures that are capable of supporting the integration of multiple types of agents and technologies [11]. Finally, decentralized agentic AI orchestration is expected to revolutionize the ability to improve both the dependability and adaptability of distributed cyber-physical systems. The use of autonomous intelligence, collaborative decision making, and adaptive coordination would provide the ability for systems to self-repair and maintain resilience to ever-changing and complex challenges. Continued advancement in study in this area would be essential in providing a key building block to form the basis of the next generation of intelligent, autonomous, and resilient infrastructure for CPS [12].

This study aims to clearly specify all aspects of Decentralized Agentic AI Orchestration in Cyber-Physical Systems. The paper is structured as follows: The first section provides an introduction to the fundamental concepts and discusses the significance of what has been proposed [13]. The second section is an overview of the existing literature relating to Decentralized AI, Multi-Agent Systems, and CPS (Cyber-Physical Systems) Resilience, where the study identifies some key gaps. The third section provides a discussion regarding the proposed framework that describes the Architecture, Components, and the Orchestration Mechanisms. The fourth section discusses the Implementation and provides Performance Evaluation examples, and/or scenarios, and/or case studies. The fifth section addresses Challenges, Limitations, and Future Research Directions. Finally, the sixth section concludes the paper with a brief summary of the key findings and contributions [14].

2. Literature Review

Recent literature on the topic of decentralized agentic AI and Cyber-Physical Systems (CPS) was reviewed to gain insights into the new trends, architectures, and challenges. The review was done in a systematic manner, involving the examination of the studies published between 2024 and 2026, with themes such as decentralized intelligence, agentic orchestration, resilience, and self-healing capabilities. The chosen works as a collective gave more emphasis on the shift from the models of central control to the model of distributed, autonomous, and intelligent controls. Further, the focus was made on how agentic AI facilitated the ability to make adaptive decisions within a complex and dynamic environment, coordinate in real-time, and enhance system robustness.

The study by Hashash et al. (2026) [15] analyzed the shortcomings of the current AI-native wireless and suggested a shift towards AGI-native by introducing perception modules, world models, and action-planning modules. Researchers concentrated their work on decentralized digital twins and synchronization based on optimization to increase reasoning and flexibility in dynamic settings. Likewise, Crespo Márquez et al. [16] (2026) also proposed an event-based multi-agent model of preventive maintenance and revealed how autonomous agents managed decision processes through explainable and transparent processes. Biswas et al. (2026) [17] identified essential governance and ethical issues with the fast Implementation of decentralized agentic systems, and the necessity of flexible regulatory frameworks.

The concept of resilience. In the resilience context, Johri et al. (2026) [18] suggested a Dual-Agent Reinforcement Learning (DARL) framework that dynamically optimized power distribution systems during uncertainty by attaining much better resilience and cost-efficiency benefits. Moreover, a human-in-the-loop CPS architecture with federated and split learning was introduced by Kalaiyarasi et al. (2026) [19] to guarantee the privacy, scalability, and ethical considerations in the decentralized setup.

In addition to these publications, Hassan et al. (2026) [20] revealed how blockchain and decentralized storage could be used to guarantee the security and traceability of UAV systems, which puts more emphasis on the significance of trust in distributed CPS. As another important element of resilient systems in the future, Mandegari et al. (2025) [21] also highlighted decentralized micro-enterprises and adaptive decision-making.

Based on these premises, a number of studies addressed the combination of agentic AI and distributed intelligence and resilience systems. Fernandez-Miguel et al. (2025) [22] also revealed the effectiveness of agentic AI-based predictive maintenance systems in enhancing accuracy and minimizing downtimes in smart manufacturing with the help of self-organizing agent ecosystems. Oyebode et al. (2025) [23] suggested a single structure integrating the models of causation, federated learning, and cryptographic proofs in order to allow secure and scalable decentralized coordination. Similarly, Farahani et al. (2025) [24] introduced a layered architecture combining the large language model-based agents with the conventional multi-agent system to make adaptive and explainable decisions. The article by Chergui et al. (2025) [25] emphasized the significance of reasoning-informed agentic AI in enhancing the performance of a network towards greater autonomy and mitigating decision biases related to cognitive biases.

Previous background knowledge was given by Zhang et al. (2024) [26], who compared the power and weakness of large language model-based agents regarding perception, reasoning (and implementation) processes. Similarly, Donsante et al. (2024) [27] highlighted the importance of model predictive control in increasing CPS adaptability and resilience in various areas, whereas Molhem et al. (2024) [28] concentrated on bringing better performances to CPS with adaptive sensor sampling methods. Together, the above studies suggested a significant evolution of systems towards decentralized, intelligent, and self-adaptive systems, as well as uncovered critical gaps in the study of issues of scalability, coordination, governance, and self-healing in real-time. The study of an interface of a multi-agent human-AI system, Nazmunisha et al. (2026) [29], suggested an autonomous system that can detect anomalies, analyze the root cause, and organize the recovery process in the hybrid edge-cloud environment. It used low-code visual control over humans, neuro-symbolic explainable reasoning, and zero-trust orchestration. Experimental findings revealed a 45 percent mean time to recovery, 30 percent lower latency, and 99.8 percent uptime, which was better than the traditional systems.

The study indicates that low-code multi-agent frameworks enhance resilience and availability in Industry 5.0 web services. The review highlights a shift from reactive monitoring to proactive, autonomous systems integrating multi-agent intelligence, human-AI collaboration, and low-code platforms. However, existing studies lack a holistic approach to real-time orchestration and self-healing, with limitations in scalability, interoperability, and user-centric design. Gaps also exist in secure coordination and ethical governance, emphasizing the need for scalable, secure, and user-friendly solutions for autonomous resilience.

3. Research Methodology

The research methodology follows a hybrid methodology, which integrates data-driven modeling, a multi-agent system, and validation by using simulation. A time-series anomaly detection dataset is processed by cleaning, normalization, and structuring. The CPS is considered to be a distributed network in which autonomous agents are assumed to be located in each node. These agents recognize anomalies with the help of CNN, RNN, and hybrid models and organize them with the help of decentralized orchestration. Agents perform self-healing and diagnostic procedures to identify faults. System performance is checked by simulation in different conditions of failure.

3.1. Dataset Used

The dataset of Anomaly Detection that is presented in Kaggle is a time-series-oriented dataset [30] that is required to detect anomalous behavior or when a system is functioning in a manner that is not normal, and this is fundamental in the development of smart systems to monitor. It is usually filled with sequential data that indicates system activities, with normal and abnormal cases being marked or discernible, allowing the creation of machine learning and deep learning frameworks to spot anomalies.

The dataset is applaudably applicable in systems that are Cyber-Physical (CPS), since it can be useful in detecting abnormal conditions like system failure, cyber-attacks, or system performance decline. The dataset can also be applied to autonomous agents to be trained to detect abnormal conditions in distributed systems and pursue relevant self-healing or recovery measures, thus making the system more resilient and reliable in the context of decentralized agentic AI coordination.

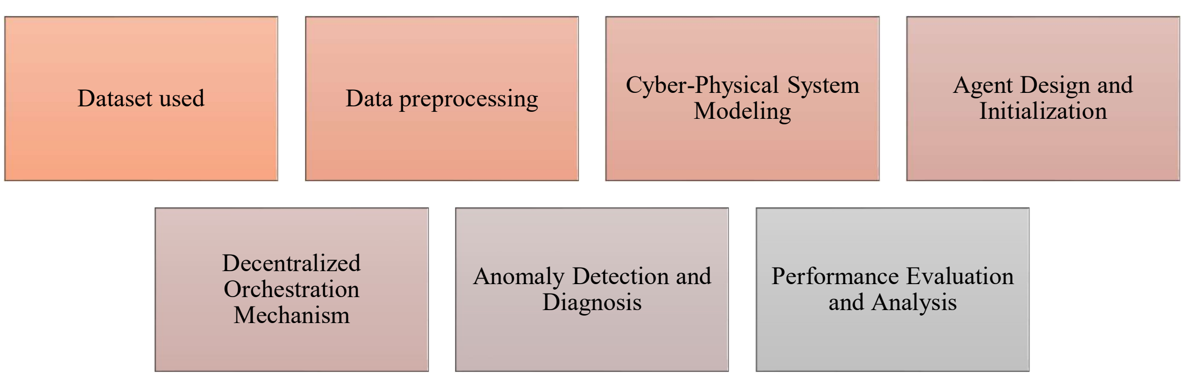

Fig. 1 Framework of Proposed Methodology

Table 1. Training–Testing Data Split Table

Dataset Component	Description	Number of Samples	Percentage (%)
Training Set	Normal system behavior used to train the model	70,000 (approx.)	70%
Testing Set	Contains both normal and anomalous data for evaluation	30,000 (approx.)	30%
Normal Data (Test)	Expected system behavior	21,000	70% of the test
Anomalous Data (Test)	Faults/attacks/deviations	9,000	30% of the test

3.2. Data Preprocessing

3.2.1. Data Cleaning and Missing Value Handling

The data is then tested for gaps, blank or inconsistent data, which could influence the model performance. The missing values are addressed with the help of statistical imputation methods, such as the mean or median substitution. To impute a feature xx, the missing values are imputed by means:

$$x_i = \begin{cases} x_i & if\ x_i \neq NaN \\ \frac{1}{N}\sum_{j=1}^{N} x_j & if\ x_i = NaN \end{cases} \quad (1)$$

This guarantees the completeness of data and eliminates bias in training.

3.2.2. Noise Filtering and Outlier Treatment

The noise and extreme outliers are detected and addressed to enhance the quality of data. Outliers are detected by using statistical tools like the Z-score:

$$Z = \frac{x-\mu}{\sigma} \quad (2)$$

μ is the mean and σ is the standard deviation. The data that contains $|Z|>3$ are treated as outliers and eliminated or limited to ensure stability in the learning of the model.

3.2.3. Data Normalization / Feature Scaling

As the data set has features of different scales, it is normalized to move all the data to a similar range. It is usually normalized by Min-Max:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (3)$$

This normalization of features makes them within the range [0,1], which enhances the convergence of machine learning models and makes the features contribute equally.

3.2.4. Time-Series Structuring (Windowing)

To detect sequential anomaly, the data is converted to time windows in order to record the temporal dependencies. It is implemented using a sliding window:

$$X_t = \{x_t, x_{t+1}, \dots, x_{t+w-1}\} \quad (4)$$

where w (so size of windows). This enables the model to acquire patterns in the long term as opposed to data elements.

3.2.5. Feature Selection and Dimensionality Reduction

Unimportant or unnecessary characteristics are eliminated in order to enhance computing power and model precision. One of the techniques is the variance thresholding or Principal Component Analysis (PCA). PCA transforms features as:

$$Z = XW \quad (5)$$

In which X represents the original feature matrix, and W is an array of principal components. This decreases the degree of dimensions and retains the highest degree of variance.

3.2.6. Label Encoding / Data Annotation

In case the dataset contains categorical labels (e.g., normal vs anomaly), it is coded to a numerical representation so that it fits the models:

$$y = \begin{cases} 0, & normal \\ 1, & anomaly \end{cases} \quad (6)$$

This is done to make sure that the dataset is appropriate to use with supervised or semi-supervised learning models.

3.2.7. Train–Test Splitting

The processed data set is separated into a training and a test data set. In the case of anomaly detection, only normal data is usually trained:

$$D = D_{train} \cup D_{test}, \quad D_{train}\ D_{test} = \emptyset \quad (7)$$

This is usually done in a 70:30 division where the model is taught normal behaviors using the training data, and it identifies unusual behaviors during the testing stage.

3.3. Cyber-Physical System Modeling

The Cyber-Physical System (CPS) is defined to be an interconnected and distributed network of heterogeneous nodes, with each node being a combination of sensing, computation, communication, and actuation functions. A graph structure is used to represent the system, where a physical device is represented as a node and a communication link as an edge [31]. Every node constantly feeds back on its environment and thus, real-time monitoring and control are possible. The local states are updated dynamically as a result of incoming data and the choices of the agents. This decentralized modeling system enables scalability, resiliency to faults, and adaptive behavior, and is the basis of applying agent-based coordination and self-healing behavior in complex CPS scenarios [32].

3.4. Agent Design and Initialization

Each CPS node has autonomous agents deployed so that it can make decentralized decisions and control. An agent is described as a state. S_t , a set of actions A_t Moreover, a policy $\pi(a \mid s)$ determines action upon observed conditions of the system. The goal is to maximize a reward function. R_t, which is a measure of system stability and resilience [33]. Learning models are used to initialize agents in which the policy is updated as:

$$\pi^* (s) = \frac{arg \max \mathbb{E}[R_t]}{a} \quad (8)$$

This model enables the agents to conform dynamically and identify anomalies as well as coordinate with the adjacent agents to heal themselves in a distributed environment efficiently.

3.5. Decentralized Orchestration Mechanism

A decentralized orchestration mechanism provides autonomous coordination of distributed agents without the use of a central controller. Agents exchange messages by passing peer-to-peer messages and exchange local state information in order to accomplish system goals on a global scale [34]. Decision-making is done through consensus or negotiation strategies, and the global state can be modeled as $\mathrm{S} = \cup_{\mathrm{i}=1}^{\mathrm{N}} \; \mathrm{S_i}$ Task assignment and recovery are dynamically coordinated, which allows the system to operate efficiently and fault-tolerantly in dynamic cyber-physical systems.

3.6. Anomaly Detection and Diagnosis

Diagnosis and anomaly detection are needed to detect the cyber-physical systems disruptions [35]. A local observation $\mathrm{x_t}$ is monitored by each agent and compared with the anticipated behavior $\hat{x_t}$. An abnormality is observed when the deviation goes beyond a level:

$$|x_t - \hat{x_t}| > \delta \quad (9)$$

A Z-score can give the statistical detection as well:

$$Z = \frac{x_t - \mu}{\sigma} \quad (10)$$

Where μ and σ are the mean and standard deviation. To diagnose it, the likelihood of the type of fault f_i is estimated as:

$$\mathrm{p}\,(f_i|x) = \frac{\mathrm{p}\,(x|f_i)\,\mathrm{p}\,(f_i)}{\mathrm{p}(x)} \quad (11)$$

This allows agents to detect root causes correctly and take the right self-healing measures.

3.6.1. CNN Model

A Convolutional Neural Network (CNN) is an architecture typically used for the extraction of relevant features from unstructured data sources such as time series signals or network traffic. This study implemented the CNN model in order to identify patterns associated with both normal and abnormal functional characteristics of Cyber-Physical Systems. The CNN architecture has three major components, including convolutional layers for feature extraction, pooling layers for dimensionality reduction, and fully connected layers for classification [36]. The CNN uses input data to be processed through multiple filters (32-128), various kernel sizes (3x3), and multiple epochs (50-100) with different batch sizes of 32 or 64 in order to achieve hierarchically learned features that facilitate more accurate anomaly detection within the system, thereby assisting multiple intelligent agents in fault identification within a system to improve system resiliency.

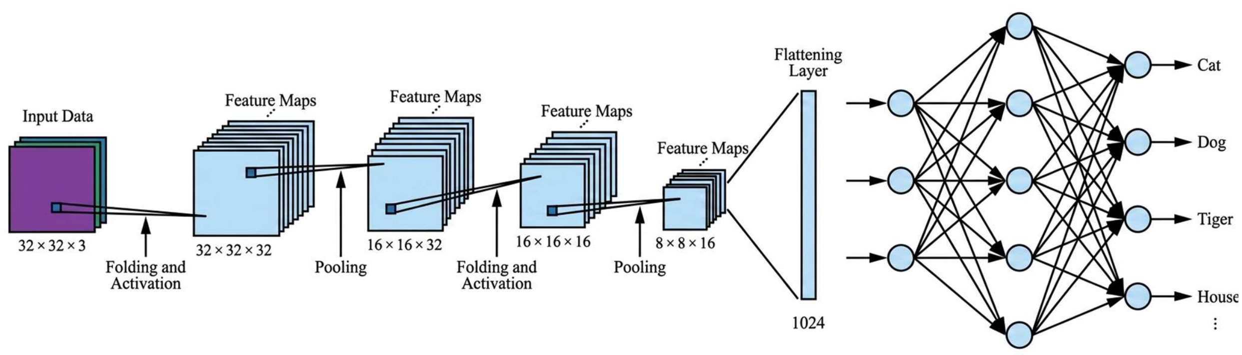

Fig. 2 CNN architecture showing convolution, pooling, and classification layers [37]

3.6.2. RNN Model

A Recurrent Neural Network (RNN) is a type of deep learning model that can be used to process sequential or time-series data with temporal dependencies between items in order to recognize patterns over time. The RNN model was used for this analysis of the continuous data that was generated by the cyber-physical systems, which include sensor readings and network traffic. RNNs are unlike most traditional models because they have some internal memory, so they can “remember” past sequences and learn from them to detect anomalies as they happen over time. In terms of the numeric configuration of our RNN Model, the RNNs were configured with hidden units (i.e., 64-128), sequence lengths (10-50 time steps), batch sizes (32-64), and number of epochs (50-100) for training on the given data set. The above characteristics enable intelligent agents to detect abnormal patterns in an accurate manner and to facilitate timely self-healing, improving reliability and stability within the systems [38].

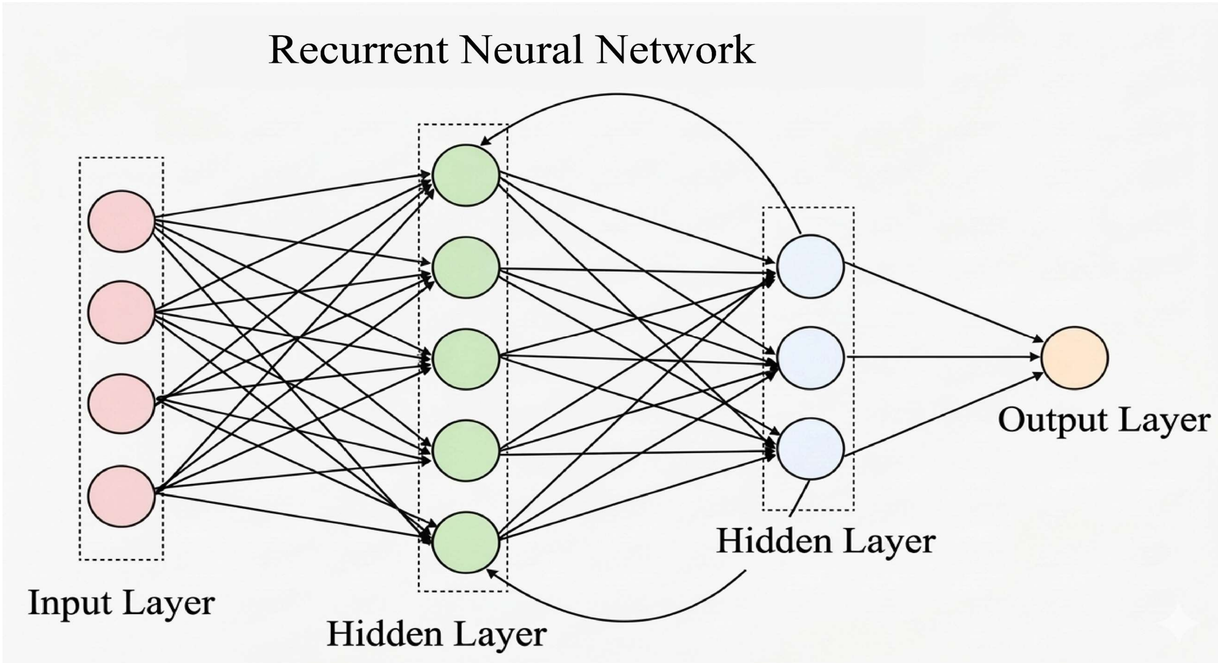

Fig. 3 RNN architecture with feedback connections [39].

3.6.3. Hybrid Model

An improved method for detecting anomalies in cyber-physical systems utilizes a hybrid approach, combining the advantages of CNN and RNN structures. In this investigation, CNN would be used to identify spatial features from the input data by applying a variety of filters (i.e., 32-64 filters each having kernel sizes of 3x3), while RNN would capture temporal dependencies using between 64 and 128 hidden units with a sequence length of between 10 and 50 timesteps.

In addition, the model would be trained using either 32 or 64 batch sizes over 50-100 epochs. This method allows for detecting complex and dynamic anomalies much more efficiently than using either model alone, reduces false positives, and provides intelligent agents with enhanced decision-making capability for autonomous self-healing and greater system resilience.

3.7. Performance Evaluation and Analysis

To assess the performance of the suggested model, conventional measures of anomaly detection and classification are used:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (12)$$

$$Precision = \frac{TP}{TP + FP} \quad (13)$$

$$Recall = \frac{TP}{TP + FN} \quad (14)$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (15)$$

$$FPR = \frac{FP}{FP + TN} \quad (16)$$

$$MTTR = \sum_{i=1}^{N} T_{repair,}\text{i} \quad (17)$$

$$\text{System Reliability } R(t) = e^{-\lambda t} \quad (18)$$

$$Latency = T_{detection} - t_{occurence} \quad (19)$$

4. Results

4.1. Visualization

The use of visualization allows for the presentation of the outputs resulting from the spatio-temporal analytic process in an easily readable and interpretable form.

The use of illustrations (line charts) can visually depict changes and trends associated with rainfall and agricultural production (manifested as yields) over time.

Thematic maps, developed using Geographic Information Systems (GIS), provide visual evidence of spatial variety in soils, climate, and other characteristics across various locations.

Figure 4 shows the trends of the training and validation losses produced by Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Hybrid Models over 20 epochs.

The training loss for CNN decreased initially from approximately 0.50 to 0.12; however, there was a sharp increase in the validation loss of CNN, which moved from approximately 1.10 to 5.20. This represents a rare example of overfitting in machine learning.

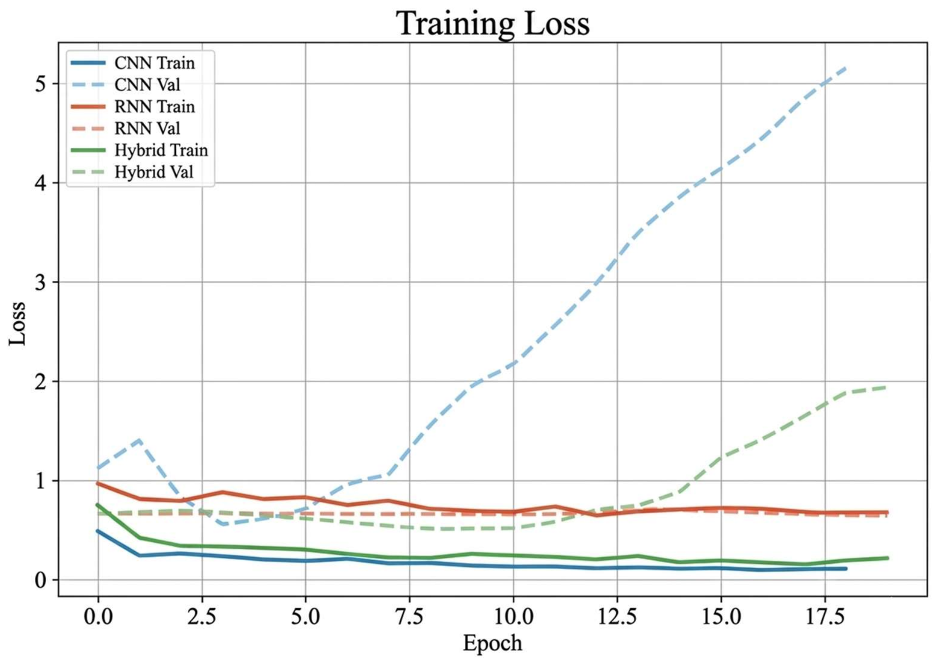

Fig. 4 Training and validation loss for CNN, RNN, and hybrid models over 20 epochs.

The losses of the RNN model decreased by a gradual amount, from approximately 0.95 to 0.68, but experienced minimal variations in validation loss, falling from approximately 0.75 to 0.70. The Hybrid Model's training loss consistently decreased over the 20 epochs, moving from approximately 0.75 to 0.22, while the Hybrid Model's validation loss initially fell to approximately 0.52 and ultimately rose to approximately 1.90; therefore, moderate overfitting in this model.

Figure 5 shows trends in accuracy across 20 epochs for the CNN, RNN, and hybrid models. The accuracy rate of the CNN training process starts from 0.78 and ends with 0.96. At the same time, the validation accuracy increases to 0.64 but quickly drops down to 0.40, pointing to a problem of overfitting. For the RNN, there is a steady improvement in accuracy rates from 0.53 to 0.62 in training. The validation accuracy remains almost constant between 0.60 and 0.66. In the case of the hybrid model, the accuracy rate of the training process ranges between 0.65 and 0.94. The accuracy rate of the validation process reaches 0.91 at the 8th epoch, then drops dramatically to 0.40.

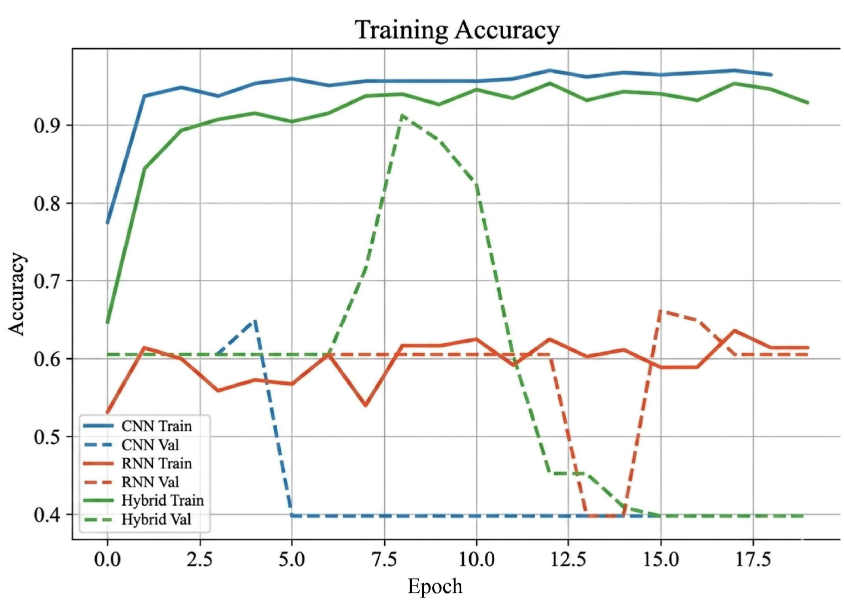

Fig. 5 Training and validation accuracy of CNN, RNN, and hybrid models.

4.2. Classification of Models

Figure 6 represents the confusion matrix for the binary classification of the CNN model. The CNN correctly identifies 72 cases of anomalies (true positives) but cannot identify any instances of the second class, i.e., there are no true negatives and no false negatives. At the same time, there are 42 samples of the normal class incorrectly classified as anomalies (false positives). As such, CNN exhibits 100% recall for the first class but demonstrates poor specificity.

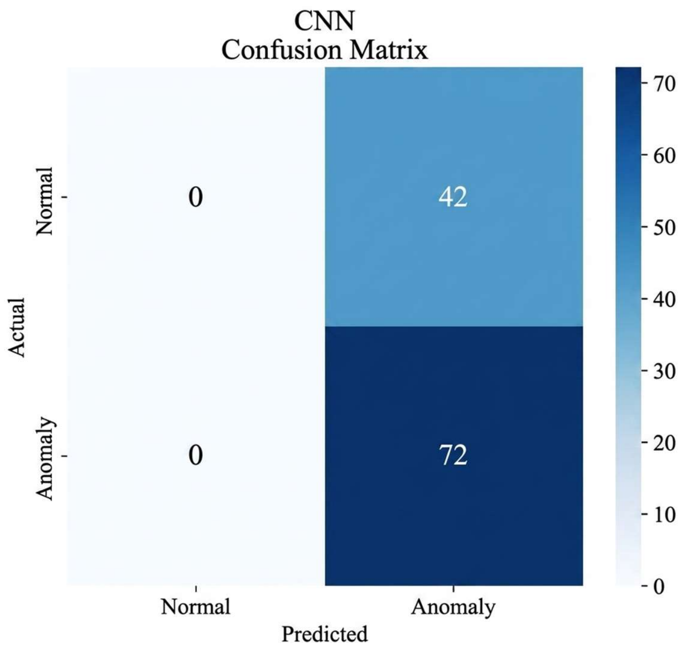

Fig. 6 CNN confusion matrix for normal and anomaly classification.

Figure 7 shows the confusion matrix for the RNN (LSTM) model. The RNN model correctly classified 72 instances of anomalies (true positives), meaning that there were no instances of normal activities identified as true negatives, and that all 42 instances of normal activities were identified incorrectly (false positives) as anomalies. There was no indication of true negatives or false negatives. This indicates that the RNN model only identified inputs as anomalies.

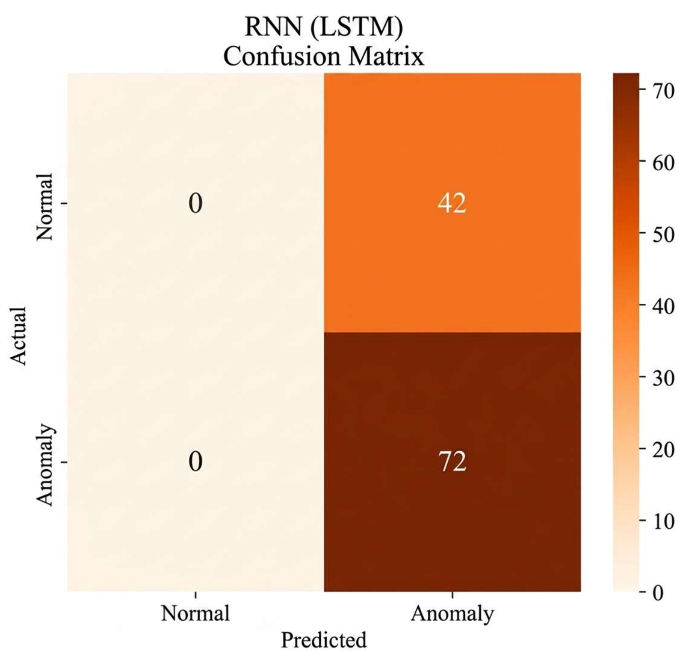

Fig. 7 Confusion matrix of the RNN (LSTM) model for normal and anomaly classification

Therefore, the RNN Model achieved a perfect Recall for the Anomaly Class (100%); however, there was no Specificity for the Normal Class. Therefore, the RNN Model demonstrates a strong bias toward the detection of anomalies (as opposed to the classification of normal activities) and results in an overall poor class balance and lack of reliability in practice.

Figure 8 shows the hybrid CNN-RNN model's confusion matrix, which shows a balanced and effective ability to classify instances accurately. The hybrid CNN-RNN model correctly classified all 40 instances of normal activity as true negatives and also successfully classified 72 instances of anomaly activities as true positives.

The hybrid CNN-RNN model incorrectly classified only 2 instances of normal activity as anomalies (false positives), and no instances of anomaly activity were classified incorrectly as normal (false negatives).

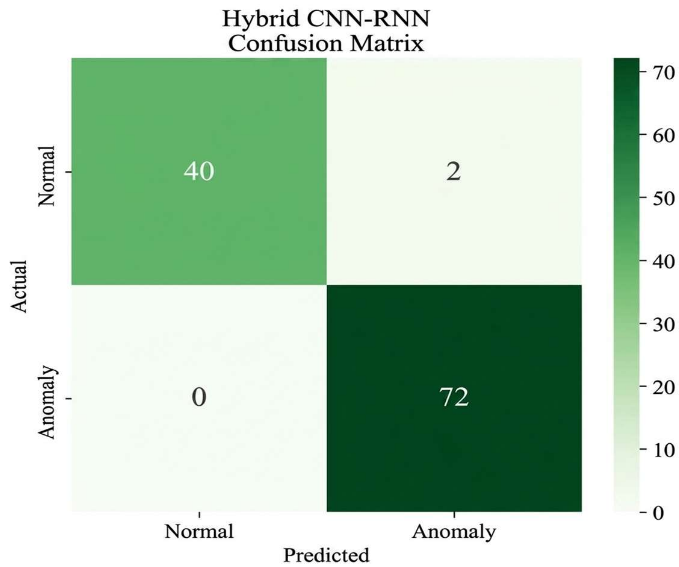

Fig. 8 Confusion matrix of the hybrid CNN-RNN model for normal and anomaly classification

Figure 9 shows the performance of the three models' classification ability through the ROC curves. The AUC of 1.000 for CNN and hybrid CNN-RNN indicates that these two models have perfect discrimination between true positive rates relative to all false positive rates, as it is close to the top-left boundary of the ROC graph. Therefore, both models provide ideal sensitivity and specificity. Although there is a slight difference, the RNN (LSTM) has almost the same AUC score (0.997) as the other two models. All three models greatly outperformed the baseline random classifier (diagonal line). Based on this outcome, it can be concluded that CNN and hybrid CNN-RNN models possess superior predictive capability compared with the standalone RNN model.

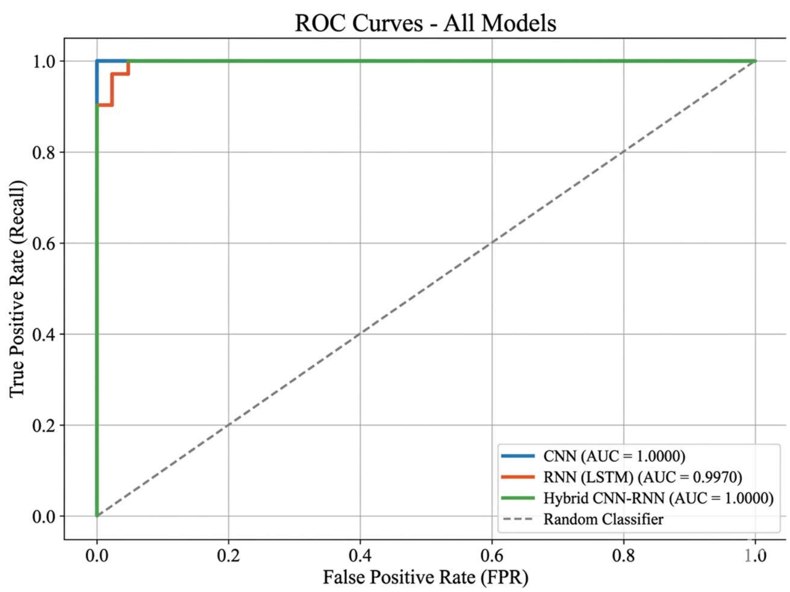

Fig. 9 ROC curves comparing CNN, RNN (LSTM), and hybrid CNN-RNN models

Figure 10 shows the performance metrics of all three models in comparison to each other. Both CNN and RNN had identical values for accuracy and precision at 0.632, with all recall values equal to 1.000, yielding an F1 score of 0.774. The AUC-ROC scores for both CNN (1.000) and RNN (0.997) are very high, demonstrating outstanding discrimination between classes, even though precision is lower than that of other models.

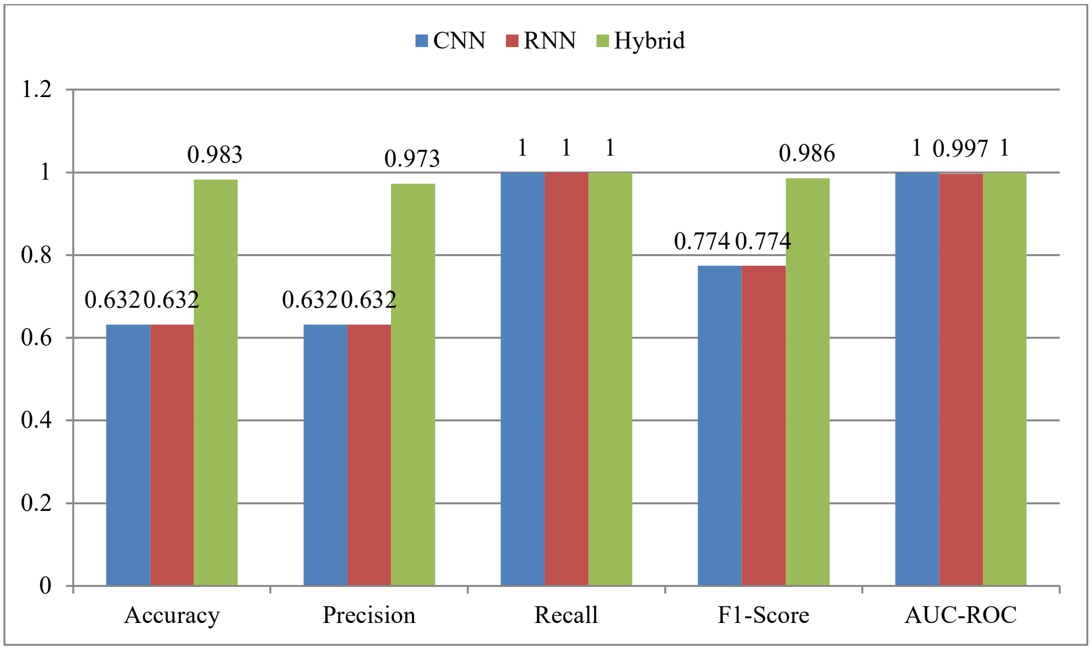

Fig. 10 Comparison of classification metrics for CNN, RNN, and hybrid CNN-RNN models

The hybrid CNN-RNN model had significantly higher performance metrics than either CNN or RNN, with an accuracy of 0.983, a precision of 0.973, a recall of 1.000, an F1 score of 0.986, and a perfect AUC-ROC of 1.000; therefore, it was the most balanced classification model.

Figure 11 shows the results of the computed reliability values for the CNN, RNN, and hybrid models using the exponential reliability function $R(t) = e^{(-\lambda t)}$. It is observed that the CNN and RNN models have similar reliability values of about 0.0025. Hence, the probability of sustaining their performance is relatively low. On the other hand, the hybrid model has a much better reliability value of 0.2899. The significant difference indicates a better level of reliability for the hybrid CNN-RNN model.

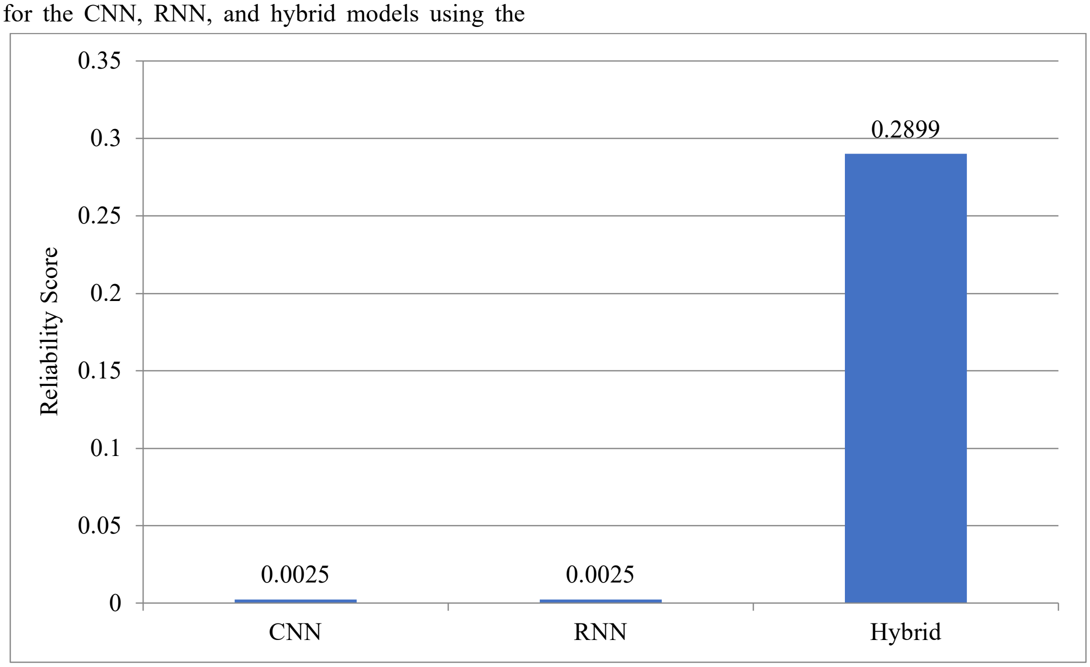

Fig. 11 System reliability scores R(t) for CNN, RNN, and hybrid CNN-RNN models

Figure 12 illustrates the detection latency values expressed by the time difference between occurrence and detection (T_detect – T_occur). The lowest latency of 10.540 ms is obtained for the CNN model. It implies that the CNN model has the shortest delay in detecting events. However, the RNN model has a higher latency of 26.750 ms, implying its slower performance in detection. The highest latency is recorded for the hybrid model, i.e., 29.860 ms, indicating increased computational complexity. Even though the hybrid model outperforms individual models in terms of accuracy and reliability, the increased latency reveals a trade-off.

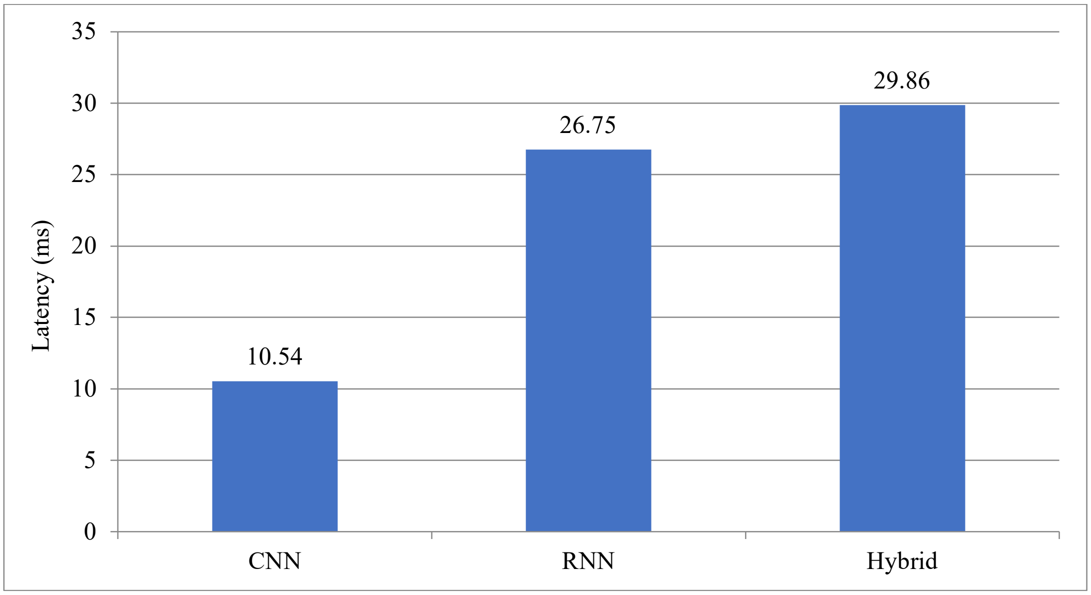

Fig. 12 Detection latency comparison for CNN, RNN, and hybrid CNN-RNN models

Figure 13 shows the values for all three different models using Equation 17 (ΣT_repair / N), the Mean Time To Repair (MTTR). The three models (the CNN, RNN, and Hybrid CNN-RNN) all have identical MTTR values of 5.02 minutes, which indicates that the systems have uniformly recovered. All three models require the same mean amount of time to restore their normal operating condition, regardless of which model is used to compute each. The MTTR for the models do not vary from one another; this indicates that the resulting relative (efficiency) model for recovery is not dependent on the type of model architecture. Overall, these results point to the equal ability to restore the system, regardless of which type of model is used (CNN, RNN, or Hybrid).

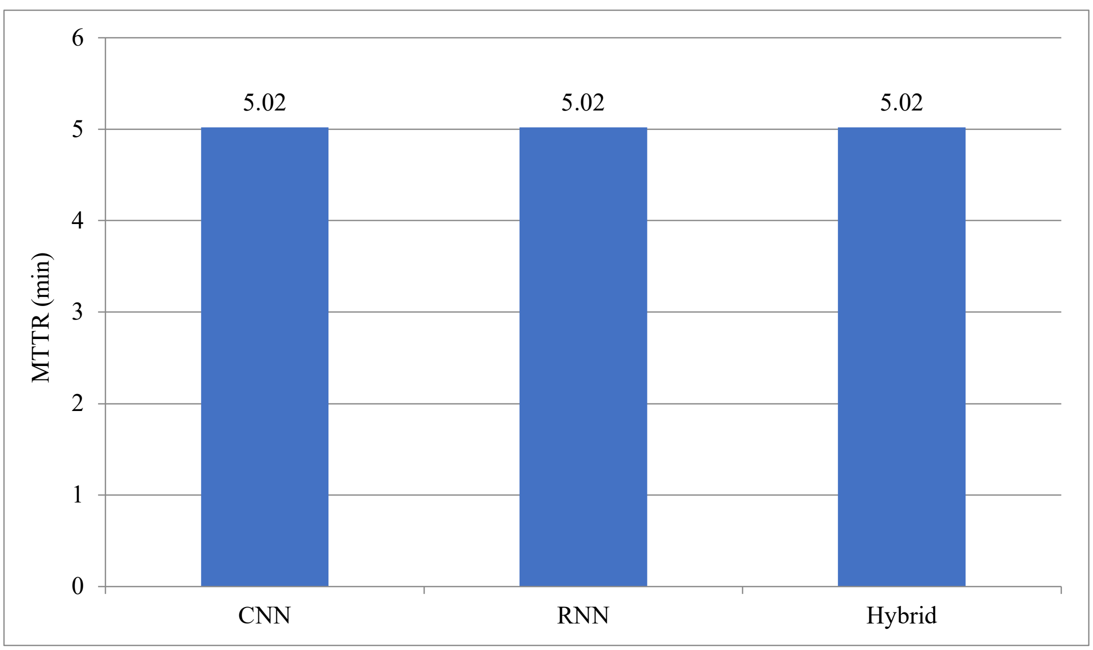

Fig. 13 Mean Time to Recovery (MTTR) comparison for CNN, RNN, and hybrid CNN-RNN models

Figure 14 shows how various models perform during the detection of abnormal events within the first 300 test samples. The CNN and RNN models (LSTM) exhibit anomaly detection capabilities throughout almost all of the period, coinciding well with the ground truth; however, these models display low selectivity of abnormal events because of their tendency to overpredict. On the contrary, the hybrid model shows higher selectivity since it falsely predicts few abnormal events. The hybrid model detects abnormalities among the first 40 samples and sustains an abnormal state thereafter.

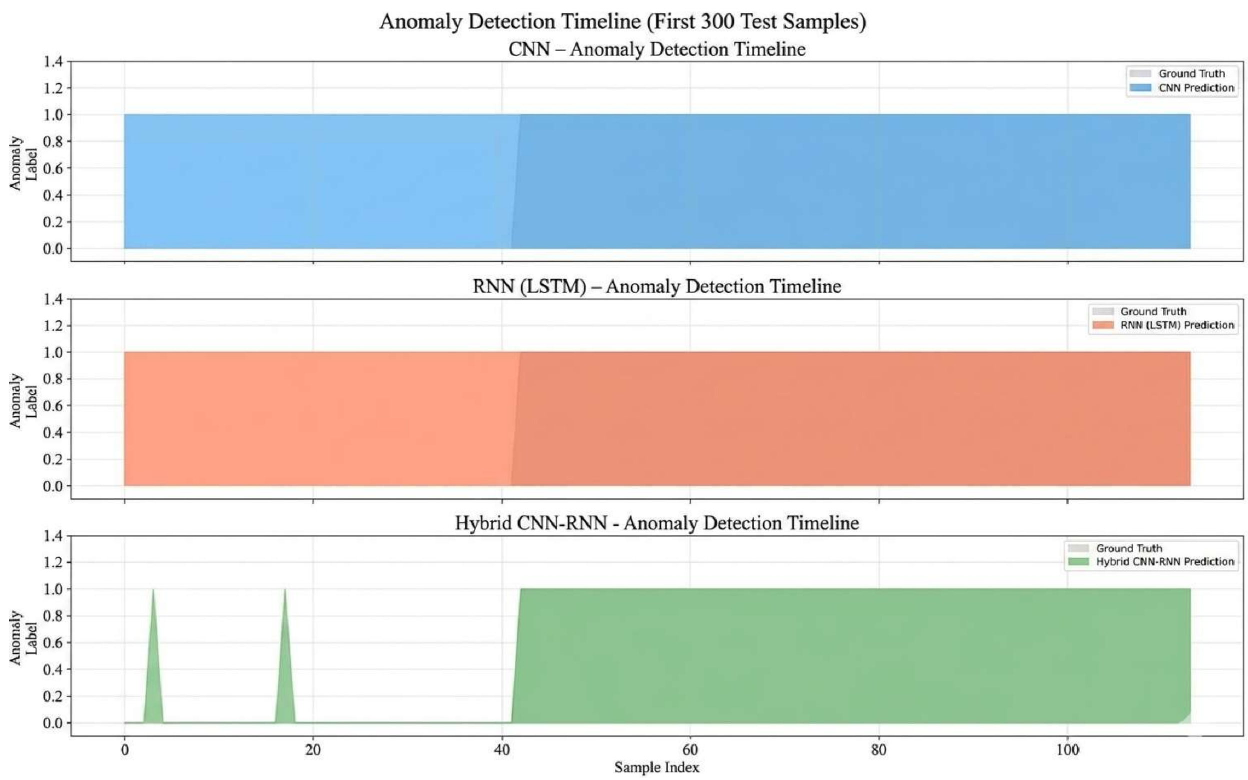

Fig. 14 Anomaly detection timelines for CNN, RNN (LSTM), and hybrid CNN-RNN models.

Figure 15 illustrates the variation of anomaly ratios across 50 time steps, with a predefined alert threshold set at 0.30. The anomaly fraction fluctuates between 0.00 and 0.60, indicating dynamic system behavior. Several peaks exceed the threshold, notably around time steps 6 (~0.60), 16 (~0.40), 18 (~0.50), and 26 (~0.50), triggering system-level alerts highlighted in red zones. For most intervals, the anomaly ratio remains below the threshold, typically ranging between 0.10 and 0.30. This pattern demonstrates intermittent bursts of anomalous activity, suggesting that the system experiences occasional instability while generally maintaining acceptable operational conditions over time.

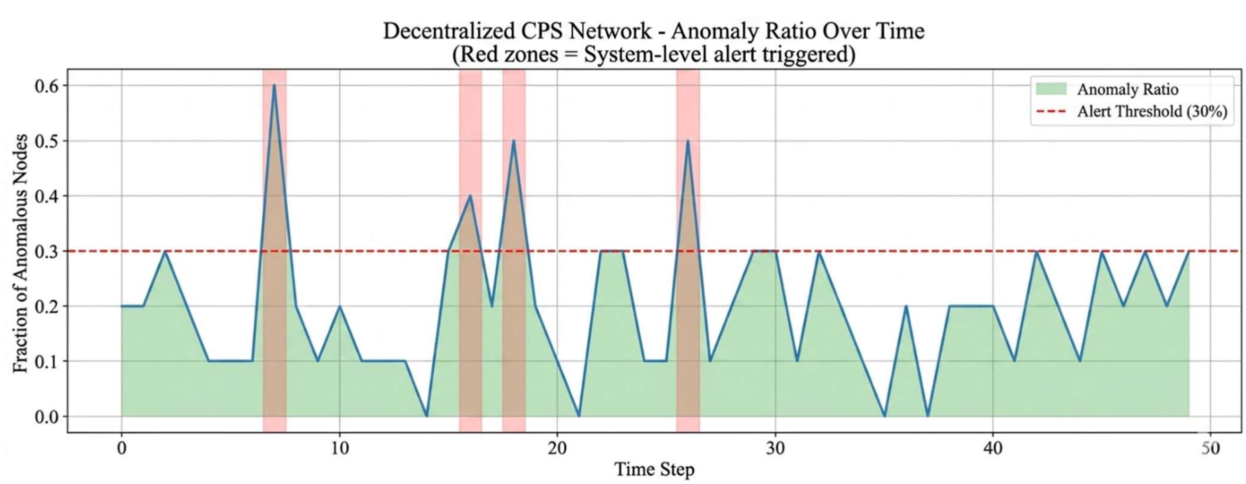

Fig. 15 Anomaly ratio over time with alert threshold in a decentralized CPS network

4.3. Decentralized Agentic AI – Full Model Comparison CNN vs RNN vs Hybrid CNN- RNN

Figure 16 provides the accuracy score for each of the three models and illustrates the differences in performance levels. The classification accuracies for both the CNN and the RNN models are the same at 0.6316, which indicates moderate classification accuracy with limited ability to differentiate between the two classification types. In contrast, the hybrid CNN-RNN model has an accuracy of 0.9825, which indicates a high level of accuracy in prediction results. Thus, the hybrid model indicates that combining CNN and RNN architectures results in utilizing both spatial and temporal capabilities in feature extraction to achieve optimal classification ability, which is therefore recommended for use in any anomaly detection threshold scenario.

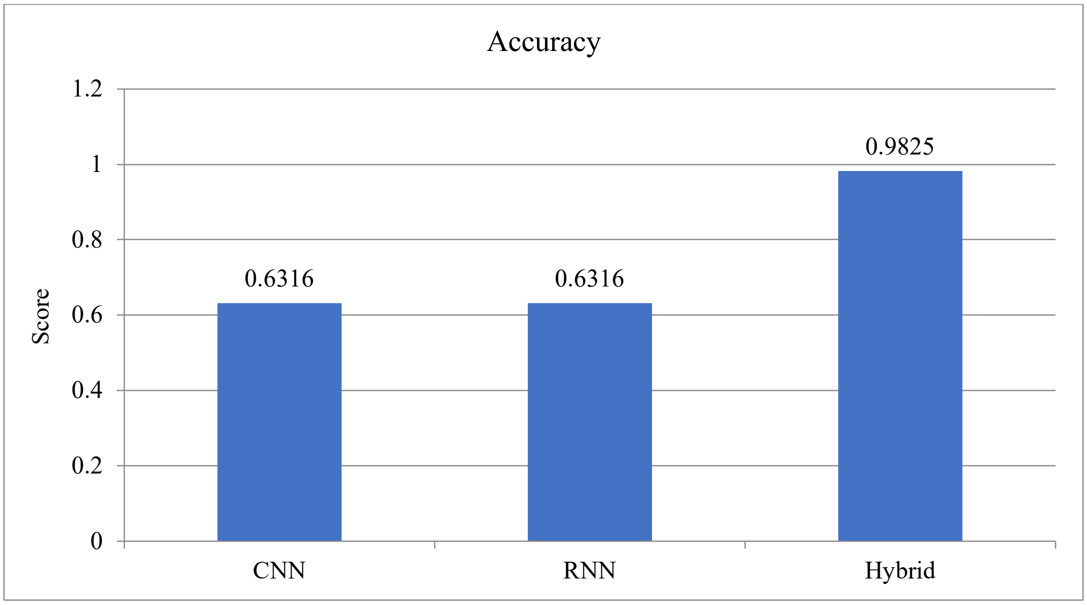

Fig. 16 Accuracy comparison of CNN, RNN, and hybrid CNN-RNN models.

Figure 17 shows the precision scores for all three models, where the differences in precision between the three models can be seen. It is evident that the CNN and RNN models attain a similar value of precision at 0.6316, which shows a fair precision in classifying true positives, although there is a prevalence of false positives. On the other hand, the hybrid CNN-RNN model shows a significant improvement in precision scores at 0.9730, showcasing its strength in identifying true positives and reducing the chances of false positives.

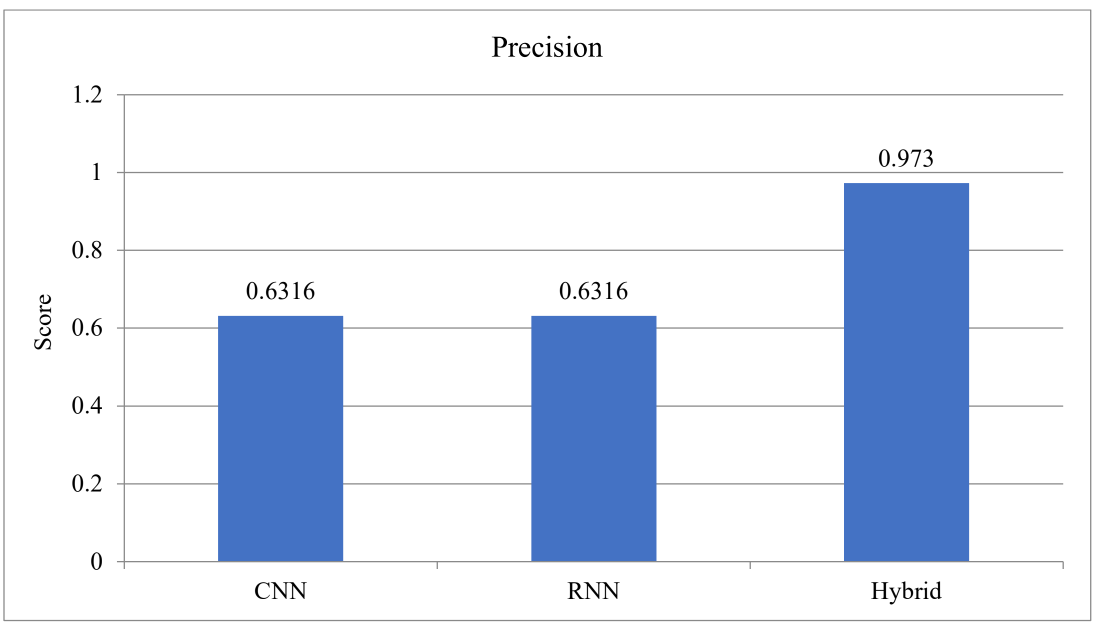

Fig. 17 Precision comparison of CNN, RNN, and hybrid CNN-RNN models

Figure 18 illustrates Recall using a bar chart format and consists of comparisons between three models (CNN, RNN, and Hybrid). The three models all show a recall score of 1.0000 or perfect sensitivity with 0% false negatives. The y-axis of the chart ranges from 0-1.00, with all three bars ending at the maximum value, and thus indicates that these three models have equal performance. All three bars have numerical labels above each bar of 1.0000; this reinforces the conclusion that none of the models exhibited any variation from one another. Therefore, all three models had equal ability to identify positive instances within the dataset. None of the models performed better than the others with respect to recall; thus, the results clearly indicate completely consistent results among all of the approaches evaluated under the conditions of the experiment.

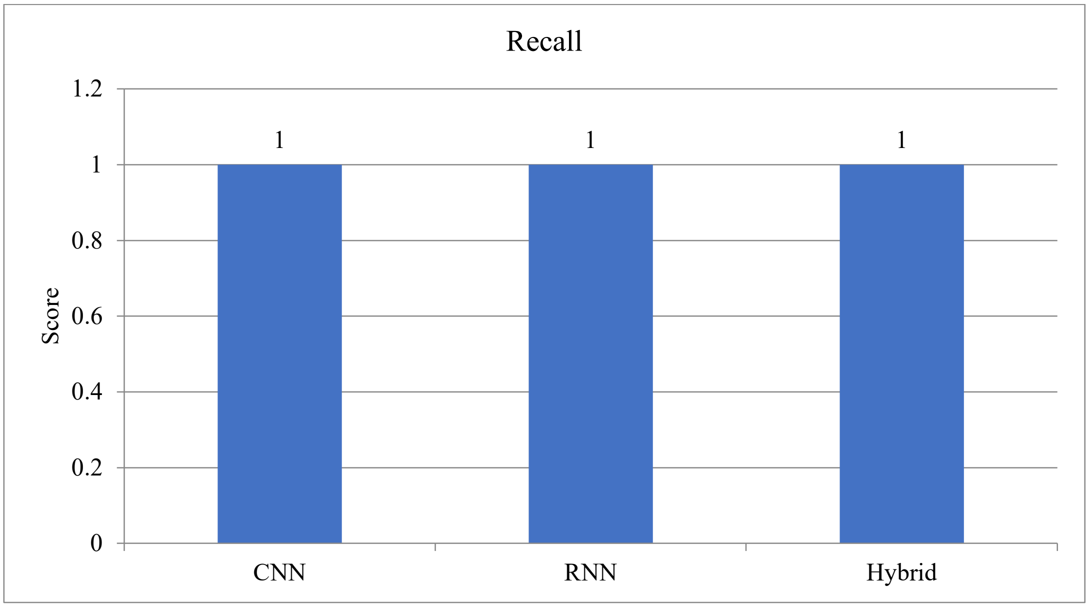

Fig. 18 All models achieve perfect Recall (1.0000).

Figure 19 shows the comparison of the F1-score between the CNN model, the RNN model, and the Hybrid model. The CNN model and the RNN model both have the same F1-score of 0.7742, which shows that both models have an average level of balance between their precision and Recall. On the other hand, the Hybrid model has a much higher F1-score of 0.9863, showing its better performance than the other two models. The difference of about 0.2121 signifies the ability of the Hybrid model to enhance the balance between classification. The y-axis values range from 0 to 1.0.

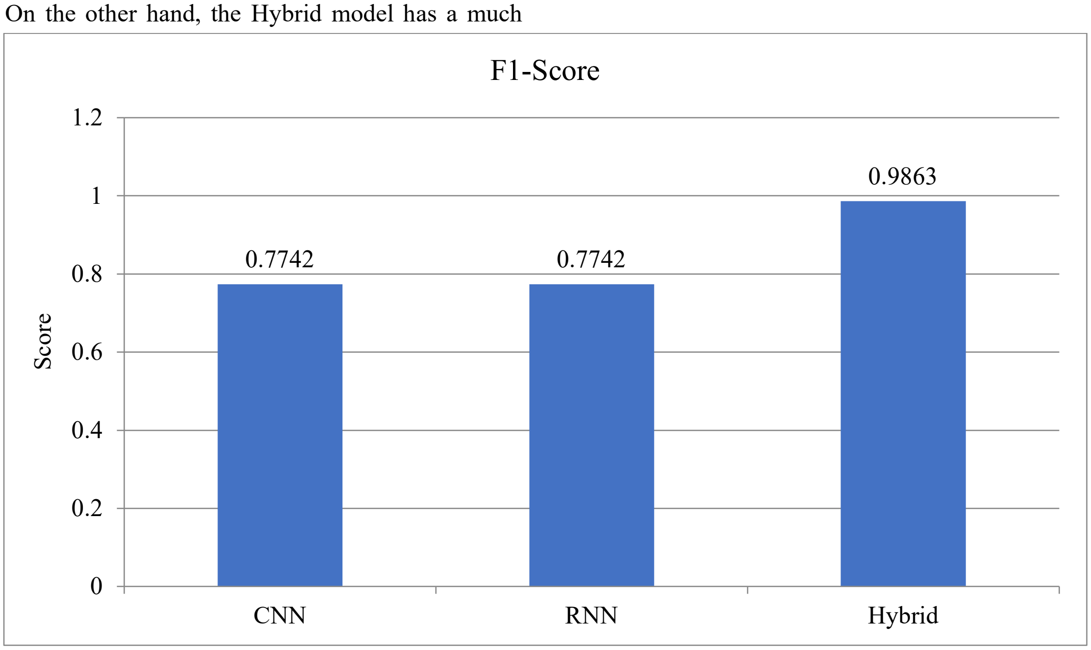

Fig. 19 F1-score comparison showing superior performance of the Hybrid model

Figure 20 shows the graph for comparison of AUC-ROC for the CNN, RNN, and Hybrid models. The two models, namely CNN and Hybrid, have attained a perfect score of 1.0000, thus depicting that they have the capability to discriminate well between the classes. RNN has attained a slightly lower score of 0.9970 compared to other models. This figure lies within a very close range of the former two models. In addition, it is 0.0030 lower than their score. The y-axis scale varies from 0.95 to 1.05. Value labels on top of the bars show the specific scores.

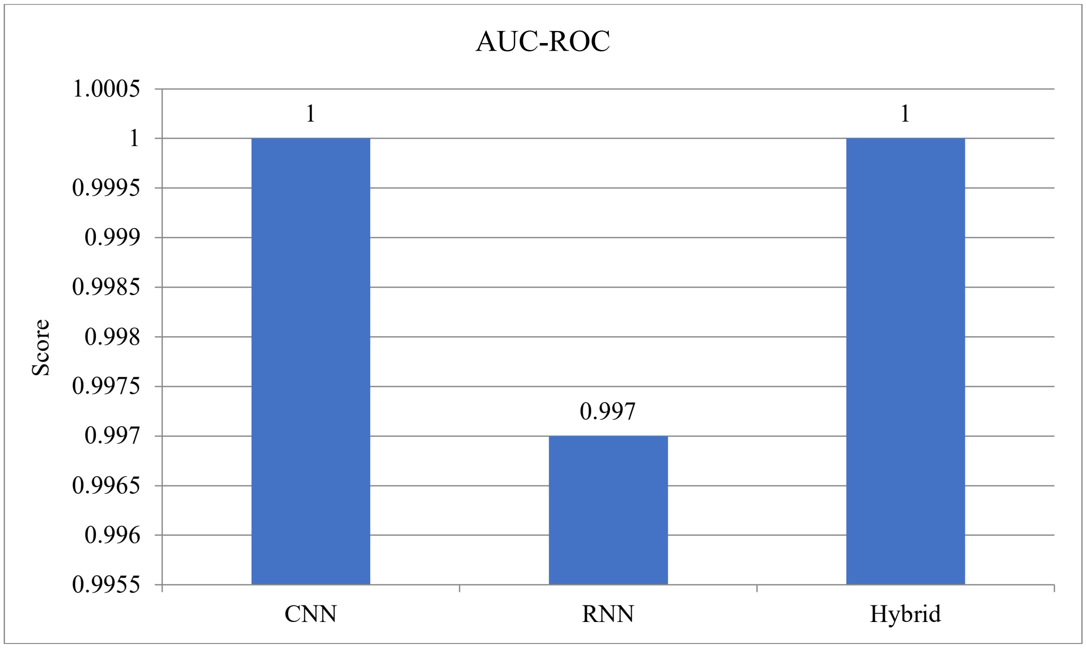

Fig. 20 AUC-ROC comparison showing near-perfect performance across all models.

Figure 21 shows the False Positive Rate (FPR) of the CNN, RNN, and Hybrid models, with lower values being associated with higher performance levels. The bar chart depicts a noticeable difference between the two models (CNN and RNN) and the hybrid one. Both CNN and RNN show high rates of FPR at 1.0000, meaning that their discrimination capability is low, and they produce many false positives. On the other hand, the Hybrid model has a significantly lower FPR of 0.0476. This shows how accurate it is compared to the other models, especially since the difference between the Hybrid model and the others is 0.9524.

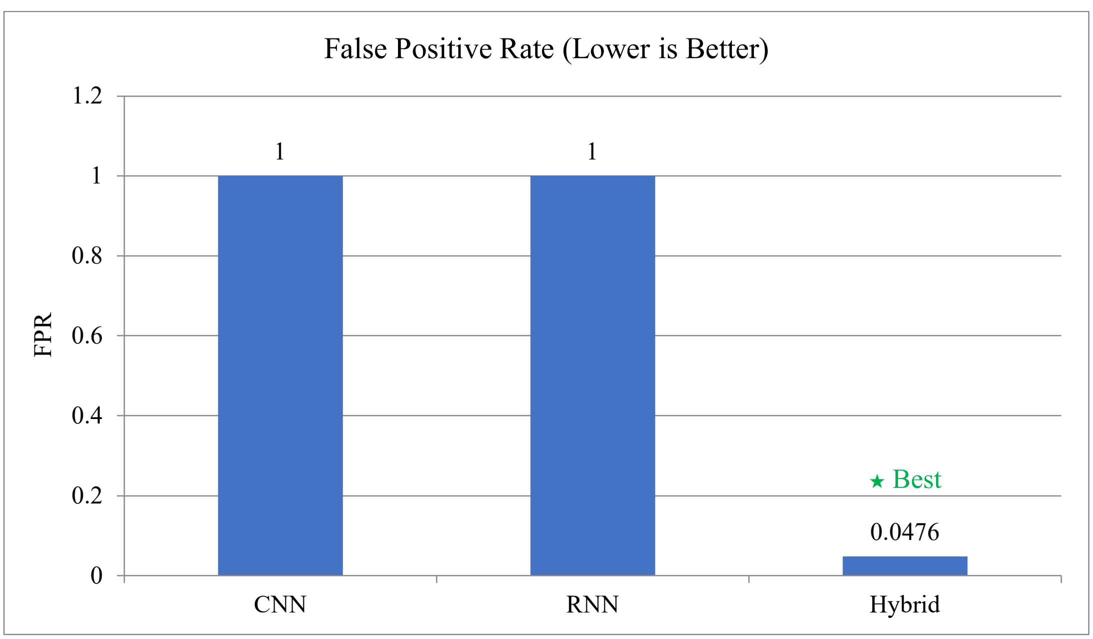

Fig. 21 False Positive Rate comparison highlighting superior performance of the Hybrid model.

Figure 22 shows the Reliability R(t) analysis of CNN, RNN, and Hybrid models. Both CNN and RNN models show extremely low reliability values, with 0.0025 being their respective reliability scores, suggesting a lack of consistency or stability over time. However, the Hybrid model is significantly more reliable with a score of 0.2899. This implies that the Hybrid model performs better and more reliably when compared to individual models. From the figure, it can be noted that the y-axis starts from 0.0 to about 0.38, and the score values are confirmed through labels.

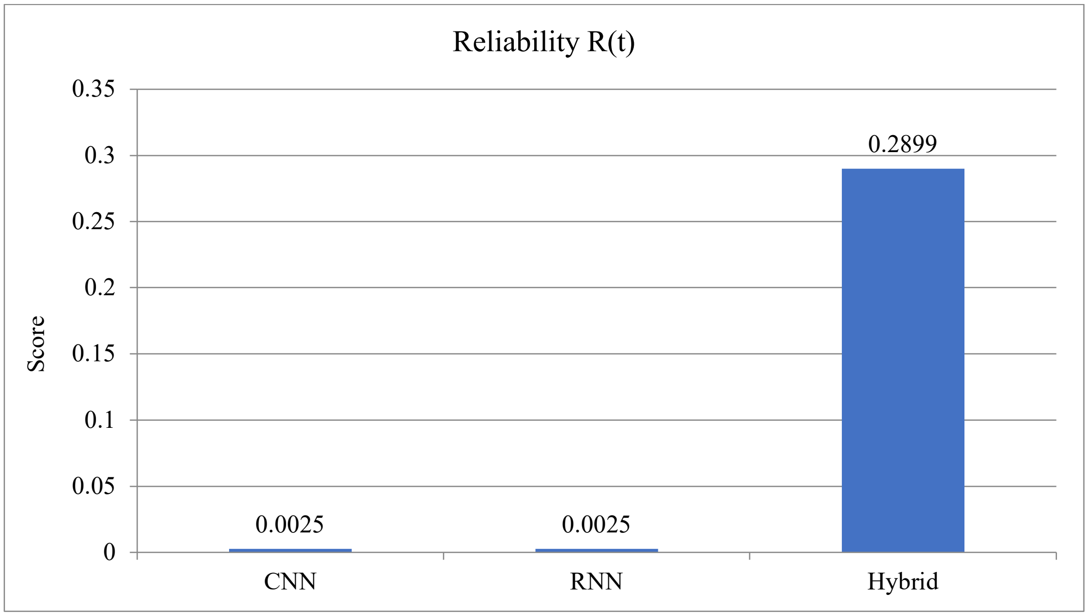

Fig. 22 Reliability comparison showing superior performance of the Hybrid model

Figure 23 below provides the MTTR comparisons between CNN, RNN, and Hybrid approaches, where smaller values represent higher efficiencies. The three models, namely, CNN, RNN, and Hybrid, have exactly the same MTTR score of 5.02 minutes. The y-axis scale varies roughly between 4.80 and 5.20 minutes, representing a limited range without any difference in values among the models. The value labels placed above the bars further confirm that all approaches take the exact recovery time. This means that there is no model superior to others regarding recovery efficiency. In general, the analysis reveals full consistency of results across all models.

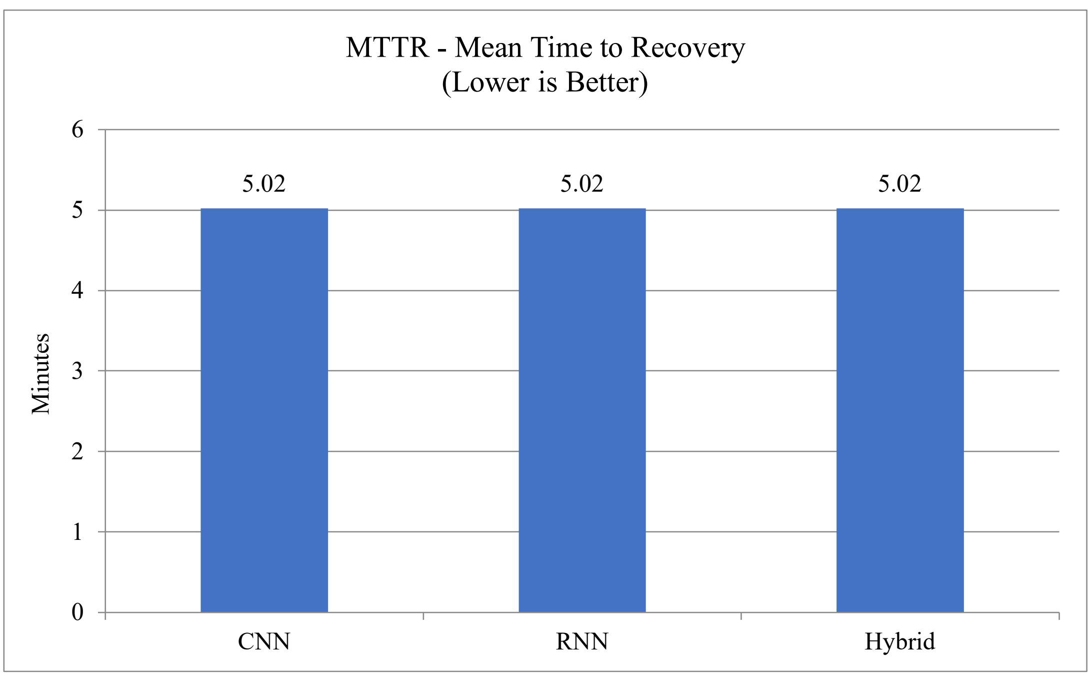

Fig. 23 MTTR comparison showing identical recovery time across all models.

Figure 24 shows the comparison of the latency time for detecting CNN, RNN, and Hybrid algorithms, where the lower number means the better result. According to the graph, the CNN algorithm detects images with the shortest latency time of 10.54 ms, which makes it the most effective among other algorithms in this regard. The RNN algorithm performs slightly worse with the latency time of 26.75 ms, whereas the Hybrid algorithm has the worst latency time of 29.86 ms. The difference between the CNN and Hybrid algorithms' latency time is 19.32 ms.

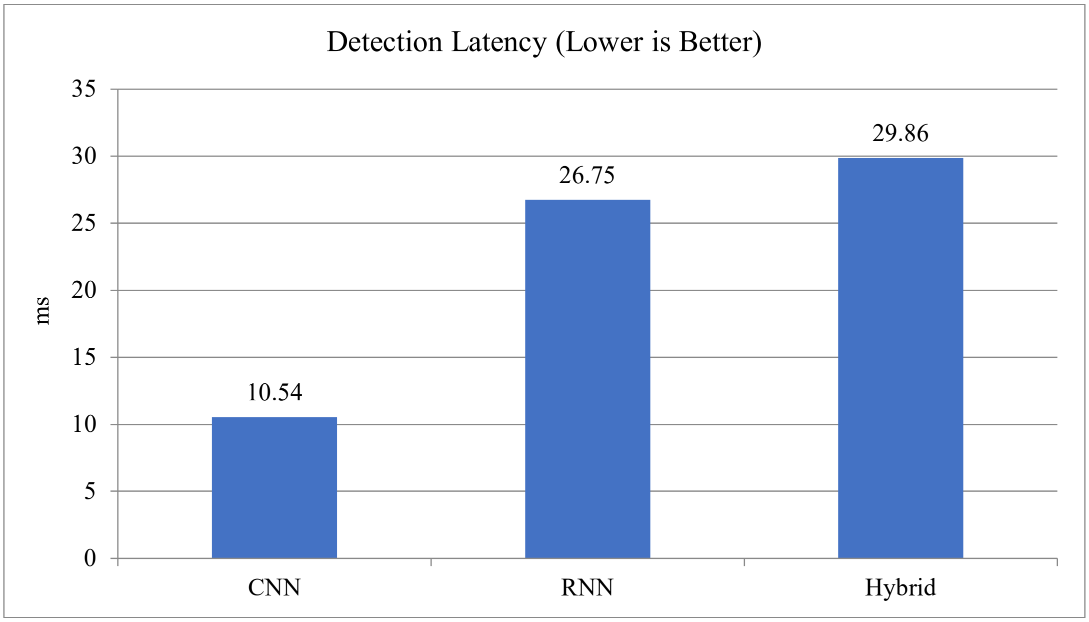

Fig. 24 Detection latency comparison showing CNN as the fastest model.

5. Conclusion

This study showcases the great potential of using decentralized agentic AI orchestration in making cyber-physical systems resilient, adaptive, and capable of self-healing processes. The proposed system avoids the problems associated with centralized approaches, such as scalability challenges, latencies, and the risk of total failures due to the presence of a single point of failure. The comparison between CNN, RNN, and CNN-RNN models reveals that even though CNN and RNN have perfect Recall, their performance is negatively affected by the high number of false positive detections and extremely poor reliability. Conversely, the CNN-RNN model demonstrates outstanding performance, which is balanced by relatively low latencies, high accuracy, precision, and F1-score. Additionally, it shows near-perfect AUC results, and most importantly, a minimal number of false positive cases and much greater reliability. It is worth noting that even though the proposed model exhibits relatively large delays, this trade-off is acceptable considering the benefits it brings compared to other approaches. Another crucial factor that the study draws attention to is the importance of incorporating both temporal and spatial learning into the process. Further research can be done on optimizing computational efficiency, increasing real-time performance, and implementing security measures and governance strategies into the system.

References

[1] Capt Noah Ugur Kilinc, "The Co-Evolution of Autonomous Production and Logistics: An Academic Synthesis of Smart Port Management, Robotic Cargo Operations, and Organic Integrity in the Era of Industry 5.0," 2026. [Google Scholar]

[2] Nicoletta González-Cancelas et al., "Optimization of Port Asset Management using Digital Twin and BIM/GIS in the Context of Industry 4.0: A Case Study of Spanish Ports," *Processes*, vol. 13, no. 3, pp. 1-23, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[3] Nistor Andrei, and Cezar Scarlat, "Marine Applications: The Future of Autonomous Maritime Transportation and Logistics," *Revolutionizing Earth Observation-New Technologies and Insights*, IntechOpen, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[4] Shaun Howell et al., "Towards the Next Generation of Smart Grids: Semantic and Holonic Multi-Agent Management of Distributed Energy Resources," *Renewable and Sustainable Energy Reviews*, vol. 77, pp. 193-214, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[5] Dillip K. Mishra et al., "A Review on Solid-State Transformer: A Breakthrough Technology for Future Smart Distribution Grids," *International Journal of Electrical Power & Energy Systems*, vol. 133, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[6] Athira Earath Shivadasan, and Prabhakar Karthikeyan Shanmugam, "Advanced Fault Detection Methodologies and Communication Protocols for DC Micro Grid - A Technical Review," *Results in Engineering*, vol. 25, pp. 1-25, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[7] Rivas, Angel Esteban Labrador, and Taufik Abrao. “Faults in Smart Grid Systems: Monitoring, Detection and Classification,” *Electric Power Systems Research*, vol. 189, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[8] Fredric Beck, and Eric Martinot, “Renewable Energy Policies and Barriers,” *Encyclopedia of Energy*, pp. 365-383, 2004. [CrossRef] [Google Scholar] [Publisher Link]

[9] Mukilan Poyyamozhi et al., “IoT—A Promising Solution to Energy Management in Smart Buildings: A Systematic Review, Applications, Barriers, and Future Scope,” *Buildings*, vol. 14, no. 11, pp. 1-31, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[10] Aitzaz Ahmed Murtaza et al., “Paradigm Shift For Predictive Maintenance And Condition Monitoring From Industry 4.0 To Industry 5.0: A Systematic Review, Challenges and Case Study,” *Results in Engineering*, vol. 24, pp. 1-24, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[11] Fatma Aktas et al., “Routing Challenges and Enabling Technologies for 6G–Satellite Network Integration: Toward Seamless Global Connectivity,” *Technologies*, vol. 13, no. 6, pp. 1-42, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[12] Md. Tonmoy Hossain et al., “Next Generation Power Inverter for Grid Resilience: Technology Review,” *Heliyon*, vol. 10, no. 21, pp. 1-40, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Fredrik Ege Abrahamsen, Yun Ai, and Michael Cheffena, “Communication Technologies for Smart Grid: A Comprehensive Survey,” *Sensors*, vol. 21, no. 23, pp. 1-24, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[14] J. Andrew et al., “Blockchain for Healthcare Systems: Architecture, Security Challenges, Trends and Future Directions,” *Journal of Network and Computer Applications*, vol. 215, pp. 1-36, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[15] Omar Hashash, *Artificial General Intelligence (AGI)-Native Wireless Systems: Digital Twins and World Models for Beyond 6G Networks*, Doctoral Dissertations, 2026. [Google Scholar] [Publisher Link]

[16] Adolfo Crespo Márquez, and Juan Francisco Gómez Fernández. “Agentic AI for Autonomous Preventive Maintenance Policy Governance: A Multi-Agent Framework for Dynamic Industrial Environments,” *Expert Systems with Applications*, vol. 314, pp. 1-19, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[17] Birupaksha Biswas, and Suhena Sarkar, “Responsible Agentic Artificial Intelligence Governance: Risk, Safety, and Ethical Challenges in Autonomous Systems,” *International Journal of Applied Resilience and Sustainability*, vol. 2, no. 2, pp. 142-167, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[18] Aniket Johri et al., “Agentic-AI Based Mathematical Framework for Commercialization of Energy Resilience in Electrical Distribution System Planning and Operation,” *Sustainable Energy, Grids and Networks*, vol. 46, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[19] N. Kalaiyarasi, and T.V. Gopal, “FLASH-BPH: A Framework for Scalable and Trustworthy Human-Centric Cyber-Physical Systems,” *International Journal of Early Childhood Special Education*, vol. 35, no. 1, pp. 1-17, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[20] Abduraouf Hassan et al., “Blockchain and NFT-based Digital Passports for UAV Preoperational Certification” *Internet of Things and Cyber-Physical Systems*, vol. 5, pp. 165-184, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[21] Siavash Mandegari, NSF/ASME Student Design Essay Competition — 2025 Challenges in the Design of Complex Systems in 2040. [Online]. Available: https://idetctravel.com/

[22] Andrés Fernández-Miguel et al., “Agentic AI in Smart Manufacturing: Enabling Human-Centric Predictive Maintenance Ecosystems,” *Applied Sciences*, vol. 15, no. 21, pp. 1-31, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[23] Oyegoke Oyebode, “Decentralized Neuro-Symbolic Cognitive Architectures: Integrating Federated Reasoning, Governance, and Causal Inference for Trustworthy, Resilient Artificial Intelligence,” *Global Journal of Engineering and Technology Advances*, vol. 24, no. 3, pp. 96-114, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[24] Mojtaba A. Farahani, Md. Irfan Khan, and Thorsten Wuest, “Hybrid Agentic AI and Multi-Agent Systems in Smart Manufacturing,” *arXiv preprint*, pp. 1-12, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[25] Hatim Chergui et al., “A Tutorial on Cognitive Biases in Agentic AI-Driven 6G Autonomous Networks,” *arXiv preprint*, pp. 1-25, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[26] Sida Zhang, Ruoxi Jia, and Zan Li, “Agentic AI Across Domains: A Comprehensive Review of Capabilities, Applications, and Future Directions,” *Journal of Computing Innovations and Applications*, vol. 2, no. 1, pp. 86-98, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[27] Manuel Donsante, “Model Predictive Control of Cyber-Physical Systems,” *Catalogo Dei Prodotti Della Ricerca*, 2024. [Google Scholar] [Publisher Link]

[28] Molhem, Mohammed. “A Novel Adaptive Sampling Algorithm for Cyber-Physical Systems,” *International Review of Applied Sciences and Engineering*, vol. 15, no. 2, pp. 161-170, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[29] N. Nazmunisha, “Multi-Agent Human-AI Systems with Low-Code Platforms Enabling Adaptive Web Services and Real-Time Anomaly Remediation in Distributed Architectures,” 2026. [Google Scholar]

[30] Anomaly Detection. [Online]. Available: https://www.kaggle.com/datasets/ipolas/anomaly-detection

[31] Francesco Vitale et al., “Process Mining for Digital Twin Development of Industrial Cyber-Physical Systems,” *IEEE Transactions on Industrial Informatics*, vol. 21, no. 1, pp. 866-875, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[32] Sofiene Lassoued, and Andreas Schwung, “Introducing PetriRL: An Innovative Framework For JSSP Resolution Integrating Petri Nets and Event-Based Reinforcement Learning,” *Journal of Manufacturing Systems*, vol. 74, pp. 690-702, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[33] Jur Van Den Berg et al., “Reciprocal Collision Avoidance with Acceleration-Velocity Obstacles,” *IEEE International Conference on Robotics and Automation*, pp. 3475-3482, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[34] Dragoljub Pokrajac, Aleksandar Lazarevic, and Longin Jan Lateckix, “Incremental Local Outlier Detection for Data Streams,” *IEEE Symposium on Computational Intelligence and Data Mining*, pp. 504-515, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[35] Bin Zhou et al., “Beatgan: Anomalous Rhythm Detection Using Adversarially Generated Time Series,” *International Joint Conference on Artificial Intelligence*, vol. 2019, pp. 4433-4439, 2019. [Google Scholar] [Publisher Link]

[36] Mücahit Altıntaş et al., “Artificial Intelligence Agents Shaping the Next Generation 6G Network Systems,” *IEEE Access*, pp. 45977-46023, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[37] Understanding Convolutional Neural Networks. [Online]. Available: https://www.embedded.com/understanding-convolutional-neural-networks/

[38] Ege Erberk Uslu et al., “Next-Generation Management and Orchestration for 6G: Emerging Trends, Architectures, and Challenges,” *IEEE Open Journal of the Communications Society*, vol. 7, pp. 2222-2269, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[39] Recurrent Neural Networks in R. [Online]. Available: https://www.geeksforgeeks.org/r-language/recurrent-neural-networks-in-r/