INTERNATIONAL JOURNAL OF ARTIFICIAL INTELLIGENCE
(ISSN: 2692-5141)
VOL.06 ISSUE 08 (2026)

AMERICAN ACADEMIC PUBLISHER

Research Article

A Lightweight Explainable AI Framework for Real-Time Threat Detection in Industrial Internet of Things Networks

Md. Rahim Hasan[1]

[1]Department of Computer Science and Engineering, National Institute of Technology and Research, Bangladesh

check for updates

Received: 12 May 2026
Revised: 2 June 2026
Accepted: 20 July 2026
Published: 26 August 2026

Page No: 01-10

Abstract

The rapid integration of Industrial Internet of Things (IIoT) technologies has transformed industrial environments into highly interconnected cyber-physical infrastructures in which sensors, edge devices, cloud platforms, industrial controllers, and analytics services continuously exchange operational data. This connectivity improves automation and resource utilization but simultaneously enlarges the attack surface and creates stringent requirements for real-time threat detection. Conventional intrusion detection approaches often face a tension between detection accuracy, computational overhead, response latency, and interpretability. This paper proposes a lightweight explainable artificial intelligence (XAI) framework for real-time threat detection in IIoT networks. The framework conceptually combines lightweight feature processing, resource-aware machine-learning inference, distributed edge-oriented analysis, and an explanation layer designed to translate model decisions into operationally meaningful evidence. The methodology is positioned through a critical synthesis of the provided literature concerning cloud computing, serverless architectures, machine learning services, scalable data processing, Kubernetes-based infrastructures, and cost-aware computation. The analysis indicates that architectural principles such as elastic processing, distributed computation, and resource-aware execution can support scalable IIoT security analytics, while explainability is necessary for improving analyst confidence and facilitating rapid incident interpretation. The proposed framework further incorporates a hierarchical detection mechanism in which computationally inexpensive screening is followed by deeper analysis for suspicious traffic. Its principal contribution is therefore an architectural model that attempts to balance detection responsiveness, computational efficiency, scalability, and interpretability rather than optimizing detection accuracy in isolation.

Keywords: Industrial Internet of Things, Explainable AI, Intrusion Detection, Real-Time Threat Detection, Edge Computing, Lightweight Machine Learning, Cybersecurity, Serverless Computing, Anomaly Detection.

INTRODUCTION

Industrial Internet of Things networks increasingly connect sensors, machines, programmable controllers, gateways, industrial applications, and cloud services into integrated operational ecosystems. Such environments generate heterogeneous and continuous data streams that can be analyzed for operational optimization as well as cybersecurity monitoring. The scale of these data streams makes distributed processing and elastic computational architectures increasingly relevant. Large-scale data processing models demonstrate the value of distributing computational workloads

across multiple processing resources rather than relying exclusively on centralized execution (Dean and Ghemawat, 2004). Similarly, cloud computing research has emphasized the ability to dynamically provide computing resources according to workload requirements (Buyya et al., 2010).

Security monitoring in IIoT environments, however, differs from conventional enterprise intrusion detection. Industrial systems frequently operate under strict latency and availability constraints. A security mechanism that introduces substantial processing overhead may interfere with operational workloads, while a detector that produces opaque predictions can make it difficult for security personnel to determine why an alarm was generated. Consequently, an effective IIoT detection architecture should address at least four interconnected requirements: computational efficiency, real-time responsiveness, detection capability, and interpretability.

The challenge becomes more pronounced as machine-learning-based security analytics are deployed across heterogeneous computing environments. Machine-learning services available through cloud platforms demonstrate the increasing feasibility of automated model training and inference at scale (Kapoor et al., 2022). Nevertheless, transferring every security-relevant data stream to centralized cloud infrastructure can introduce latency and computational costs. Cost-aware cloud computing research emphasizes the importance of considering resource utilization when elastic services are designed (Yau and Chen, 2016). This principle is particularly relevant to IIoT networks, where thousands of devices may continuously generate telemetry.

The central problem addressed in this paper is therefore the design of a security framework that can provide rapid threat screening while limiting computational requirements and simultaneously explaining its decisions. The proposed approach draws architectural inspiration from distributed data processing, serverless execution, cloud-native infrastructure, and machine-learning platforms represented in the supplied literature. Recent work on hybrid ensemble intrusion detection for Industry 5.0 further reinforces the importance of advanced detection architectures for increasingly connected manufacturing environments (Govindarajan et al., 2026).

The objective of this research is to develop a conceptual, research-driven framework for lightweight explainable threat detection in IIoT networks. Specifically, the paper seeks to establish an architecture that separates data acquisition, lightweight preprocessing, threat classification, explanation generation, and response coordination. The scope is limited to architectural and analytical development rather than claiming results from a newly conducted experimental dataset. The significance of the work lies in treating computational efficiency and explainability as integral security requirements rather than secondary characteristics.

LITERATURE REVIEW

The supplied literature provides several architectural foundations relevant to the development of a lightweight IIoT security framework, although most of the references are concerned with cloud computing, distributed processing, serverless systems, or machine-learning infrastructure rather than IIoT intrusion detection directly. This creates both an opportunity and a research gap.

Dean and Ghemawat established the MapReduce paradigm as a mechanism for processing large datasets across clusters through distributed computation (Dean and Ghemawat, 2004). From an IIoT security perspective, the underlying principle is significant because industrial networks can generate substantial volumes of telemetry. A distributed analytical model can prevent a single processing component from becoming a bottleneck. However, direct adoption of batch-oriented processing is insufficient for

real-time threat detection because security decisions may need to be made within very short operational windows.

Buyya, Yeo, and Venugopal examined market-oriented cloud computing and the provisioning of computing resources as utilities (Buyya et al., 2010). This resource-oriented perspective supports elastic security analytics in which computational capacity can increase when network activity rises. The limitation is that cloud-centric elasticity does not automatically resolve latency requirements associated with industrial operations. A security framework therefore needs to determine which analytical tasks should remain close to the industrial network and which can be delegated to centralized infrastructure.

The literature concerning serverless computing provides another relevant architectural dimension. Chowdhury and Hill discussed the economic and architectural impact of serverless computing (Chowdhury and Hill, 2016), while Donaldson et al. examined serverless computing from a broader architectural perspective (Donaldson et al., 2021). Serverless execution can be useful for event-driven security functions because processing resources can be invoked when suspicious events occur rather than remaining continuously allocated. However, excessive dependence on remote function invocation may introduce execution overhead or unpredictable latency, making selective use more appropriate for time-sensitive IIoT environments.

Research on AWS-oriented machine learning provides a foundation for automated analytics. Kapoor, Sharma, and colleagues examined machine learning using Amazon SageMaker (Kapoor et al., 2022), while Reddy and Bhat surveyed machine-learning solutions powered by AWS (Reddy and Bhat, 2023). These works demonstrate the importance of managed machine-learning environments for scalable model development. Yet cloud-based machine learning alone does not solve the interpretability problem. A detector can be computationally efficient while still being operationally difficult to trust if its predictions cannot be interpreted.

The references concerning AWS Lambda and cloud-native infrastructure further support distributed execution. Mool chandani and Gupta examined AWS Lambda for serverless data-processing pipelines (Moolchandani and Gupta, 2022), while Singh, Choudhury, and Joshi discussed AWS and Kubernetes as a cloud-native approach for scalable and resilient systems (Singh et al., 2021). Together, these studies indicate that security analytics can benefit from modular and scalable execution. Kapoor and Kumar's work on infrastructure management with AWS CloudFormation also illustrates the importance of automated infrastructure provisioning (Kapoor and Kumar, 2023).

Rao and Patel examined real-time big-data analytics using AWS EMR and Apache Spark (Rao and Patel, 2023), directly reinforcing the relevance of scalable analytics for continuous data streams. Smith, Doe, and Rogers considered cloud cost management (Smith et al., 2023), while Xu, Chen, and Zhao investigated AWS adoption (Xu et al., 2017). These studies collectively indicate that computational architecture, resource consumption, and scalability cannot be treated independently.

The major gap is therefore evident: the supplied literature establishes strong foundations for scalable cloud and machine-learning architectures, but it provides limited direct integration of lightweight detection, IIoT constraints, and explainable security decisions. The proposed framework addresses this gap by combining these architectural principles with an explicit explanation layer. This positioning is complementary to contemporary hybrid intrusion-detection research for Industry 5.0 environments (Govindarajan et al., 2026).

METHODOLOGY

3.1 Research Design

This study adopts a conceptual research methodology based on architectural synthesis and analytical framework development. Rather than presenting fabricated experimental measurements, the methodology derives design requirements from the technical characteristics represented in the supplied references and maps those requirements to an IIoT-oriented security architecture.

The framework is organized around five functional layers: data acquisition, lightweight preprocessing, threat detection, explainability, and response orchestration. A sixth management plane provides model and infrastructure management across edge and cloud resources.

The fundamental design principle is to perform the least expensive computation as close as possible to the data source while reserving more computationally demanding analysis for suspicious events. This hierarchical design is intended to reduce unnecessary processing and communication overhead.

3.2 IIoT Data Acquisition Layer

The acquisition layer receives network packets, flow statistics, device telemetry, protocol metadata, authentication events, and operational indicators from industrial devices and gateways. Because IIoT networks can contain heterogeneous equipment, the architecture avoids dependence on a single telemetry format.

Data are converted into a normalized representation before analysis. Instead of transmitting complete raw traffic to a centralized platform, the proposed architecture prioritizes compact features such as packet rates, connection duration, protocol frequencies, source-destination relationships, request patterns, and temporal deviations.

This approach reflects the broader principle of distributed processing established by MapReduce: computational work can be partitioned rather than concentrated in one location (Dean and Ghemawat, 2004). In IIoT environments, however, partitioning is performed with stronger latency constraints.

3.3 Lightweight Feature Processing

The preprocessing component transforms raw telemetry into a compact feature vector. The objective is not simply dimensionality reduction but the removal of redundant information that unnecessarily increases inference cost.

A lightweight pipeline can include normalization, missing-value handling, categorical encoding, temporal aggregation, and feature selection. For example, a gateway may calculate the number of connection attempts generated by a device over successive time windows instead of forwarding every individual packet.

This strategy reduces bandwidth consumption and allows the first detection stage to execute on comparatively constrained edge resources. It also complements the resource-awareness principles found in cost-oriented cloud computing research (Yau and Chen, 2016).

3.4 Hierarchical Threat Detection

The proposed detection mechanism consists of two stages. The first is a lightweight screening model designed to distinguish normal behavior from potentially suspicious activity with minimal computational expense. The second stage is activated only when the first stage identifies an anomaly or uncertain event.

The first stage can employ a compact supervised classifier or anomaly-scoring mechanism. Its output is represented as a risk score:

$$R = f(X, M, T)$$

where X represents observed network features, M represents the trained detection model, and T represents temporal or contextual information.

If the resulting score exceeds a predefined threshold, the event is passed to the second analytical stage. The second stage can perform a more detailed classification using a comparatively complex model. This structure reduces the number of events subjected to expensive analysis.

The concept is compatible with hybrid detection architectures, where multiple analytical mechanisms are coordinated to improve security decisions (Govindarajan et al., 2026). The distinction in the present framework is that computational lightweightness and explanation are explicit architectural objectives.

3.5 Explain ability Layer

Explain ability is positioned as a dedicated layer rather than being treated as an optional visualization feature. For each security prediction, the system should identify the principal characteristics that influenced the decision.

For example, an alert could indicate that an industrial gateway was classified as suspicious primarily because of an abnormal connection frequency, unusual destination distribution, and deviation from its historical communication pattern. Such information is more actionable than a binary statement that an event is malicious.

The explanation mechanism can be implemented through feature-attribution techniques appropriate to the selected model. The output should remain computationally lightweight enough to avoid negating the performance benefits of the detection architecture.

The theoretical rationale is that predictive accuracy alone is insufficient in operational security. A security analyst must understand the basis of an alert before deciding whether to isolate a device, block communication, or initiate further investigation. Thus, explainability functions as a decision-support mechanism.

3.6 Edge–Cloud Resource Allocation

The proposed framework uses a hybrid edge-cloud architecture. Time-critical screening and basic explanation generation are performed near the industrial network, while model training, historical analysis, centralized policy management, and computationally intensive investigations can be delegated to cloud resources.

This allocation follows the broader cloud-computing principle of dynamically provisioning resources according to demand (Buyya et al., 2010). Managed machine-learning environments further demonstrate how training and analytical workloads can be integrated into scalable platforms (Kapoor et al., 2022).

Serverless execution can be applied selectively for event-triggered workloads, such as generating an extended investigation report after a high-risk event. However, continuously executing latency-critical detection through remote serverless functions may be inappropriate because the additional invocation and communication path can affect responsiveness (Chowdhury and Hill, 2016; Moolchandani and Gupta, 2022).

3.7 Model and Infrastructure Management

The management plane maintains model versions, thresholds, feature definitions, security policies, and deployment configurations. Automated infrastructure management principles can reduce configuration inconsistencies when multiple gateways and cloud services are involved (Kapoor and Kumar, 2023).

Kubernetes-oriented cloud-native approaches can additionally support resilient deployment of analytics components where orchestration requirements justify their operational complexity (Singh et al., 2021). Nevertheless, lightweight IIoT deployments should avoid introducing unnecessary orchestration overhead. Infrastructure complexity must remain proportional to the operational requirements of the environment.

3.8 Real-Time Response

When an event reaches a critical risk level, the response component can trigger predefined actions such as alert escalation, temporary communication restriction, device isolation, or collection of additional telemetry. The framework does not assume that every machine-learning prediction should automatically result in an operational intervention.

A confidence-aware policy can distinguish between low-risk anomalies requiring observation and high-confidence threats requiring immediate action. This reduces the possibility that false positives could unnecessarily disrupt industrial processes.

RESULTS

The analytical findings indicate that the proposed architecture addresses the principal tension between security analytics and resource limitations by distributing detection responsibilities across computational layers. The most significant finding is that lightweight detection should not be interpreted as the use of an inherently simple model alone. Instead, lightweightness emerges from the combined optimization of feature representation, processing location, inference frequency, and workload allocation.

The first finding concerns latency reduction. Continuous transmission of raw industrial traffic to centralized infrastructure can create unnecessary communication and processing overhead. By performing initial feature extraction and screening at the edge, the framework reduces the amount of information that must traverse the network. Distributed computation principles provide theoretical support for this approach (Dean and Ghemawat, 2004). The expected implication is particularly important for industrial networks in which security decisions may need to be produced without depending entirely on distant infrastructure.

The second finding concerns resource efficiency. A two-stage detection architecture avoids applying computationally expensive analysis to every event. Normal traffic can be processed using the lightweight screening layer, while suspicious events receive deeper analysis. This design is consistent with cost-aware computing principles in which computational resources should be aligned with workload requirements (Yau and Chen, 2016). The approach also complements elastic cloud architectures that dynamically allocate resources according to demand (Buyya et al., 2010).

The third finding concerns scalability. The literature indicates that cloud-native, serverless, and distributed processing architectures can support workloads whose computational requirements vary over time (Chowdhury and Hill, 2016; Donaldson et al., 2021). Applied to IIoT security, these principles allow intensive historical analysis and model management to scale independently from edge detection. However, scalability is not equivalent to unlimited centralization. The proposed framework therefore assigns latency-sensitive tasks to edge resources and elastic analytical workloads to cloud

infrastructure.

The fourth finding is improved operational interpretability. An XAI layer transforms model outputs into evidence that can be evaluated by security personnel. This can reduce the gap between machine-generated alerts and human decision-making. In a practical example, an abnormal industrial controller could be flagged because its communication frequency and destination pattern significantly differ from its established operational profile. Such an explanation allows an analyst to assess whether the behavior reflects an attack, maintenance activity, or legitimate operational change.

The fifth finding concerns architectural compatibility with contemporary hybrid detection approaches. Research on hybrid ensemble intrusion detection in Industry 5.0 demonstrates the relevance of combining multiple analytical mechanisms for complex manufacturing environments (Govindarajan et al., 2026). The proposed framework extends this conceptual direction by emphasizing lightweight execution and explainable outputs alongside detection capability.

Nevertheless, the findings should be interpreted as analytical rather than experimentally validated. The provided literature does not supply a common IIoT benchmark dataset, standardized threat taxonomy, or controlled performance measurements from the proposed architecture. Therefore, claims regarding exact accuracy, latency, energy consumption, or false-positive rates cannot responsibly be presented as empirical results. Experimental validation remains necessary before deployment.

DISCUSSION

The proposed framework contributes to the literature by reframing real-time IIoT threat detection as a multi-objective architectural problem. Traditional evaluation often prioritizes classification accuracy, but a security detector operating inside an industrial environment must also satisfy latency, resource, scalability, reliability, and interpretability requirements. A model that achieves high predictive performance while consuming excessive computational resources may be unsuitable for constrained gateways. Conversely, a highly lightweight model that generates unexplained or unreliable alerts may create operational risks.

The principal theoretical implication is that edge intelligence and cloud scalability should be complementary rather than competing paradigms. The cloud literature emphasizes elastic resource provisioning and large-scale analytics (Buyya et al., 2010; Rao and Patel, 2023), whereas the proposed architecture assigns immediate screening to edge resources. This creates a hierarchical computational model in which local processing provides responsiveness and cloud processing provides capacity. The model therefore adapts distributed computing principles to the particular constraints of industrial security.

Serverless computing introduces an important trade-off. Its event-driven execution model is attractive for irregular security workloads, particularly when an investigation is initiated only after an anomaly is detected. However, serverless execution should not automatically replace continuously available edge inference. Invocation overhead, communication dependencies, and platform coupling may reduce the suitability of serverless functions for the most latency-sensitive operations. The literature's discussion of serverless economics and architecture supports selective rather than indiscriminate adoption (Chowdhury and Hill, 2016; Donaldson et al., 2021).

Explainability introduces a second major trade-off. Generating explanations for every low-risk event can itself consume computational resources. Consequently, the framework should apply adaptive explanation policies. Routine events may receive compact

explanations, whereas high-risk events can trigger more detailed attribution and contextual analysis. This preserves the principle of lightweight operation without eliminating interpretability.

Another important issue is model drift. Industrial communication patterns can change because of equipment replacement, software updates, production schedules, or process modifications. A detector trained on historical behavior may consequently classify legitimate changes as anomalies. Cloud-based machine-learning platforms provide mechanisms for scalable model development (Kapoor et al., 2022; Reddy and Bhat, 2023), but retraining should be governed by industrial context rather than automated solely on statistical changes.

The framework also raises questions concerning infrastructure complexity. Kubernetes and automated infrastructure-management approaches provide valuable mechanisms for scalable deployment (Singh et al., 2021; Kapoor and Kumar, 2023), yet excessive infrastructure can conflict with the lightweight objective. A small industrial site may not require a full orchestration environment. Architectural selection should therefore be based on workload scale, resilience requirements, and administrative capacity.

Cost is similarly relevant. Cloud processing can provide flexible capacity, but continuous high-volume telemetry transmission and analysis may create substantial expenditure. Research concerning cloud cost optimization and AWS adoption indicates that economic considerations are inseparable from scalable computing architectures (Smith et al., 2023; Xu et al., 2017). The proposed edge-first design can reduce unnecessary data transfer while preserving cloud resources for tasks that genuinely benefit from centralized computation.

Finally, the framework should be positioned alongside contemporary hybrid intrusion-detection research rather than considered a replacement for it. The Industry 5.0-oriented Aegis-5 framework demonstrates the continuing development of sophisticated ensemble-based intrusion detection (Govindarajan et al., 2026). The present work emphasizes a different optimization objective: making security intelligence operationally feasible for resource-constrained, latency-sensitive environments while preserving explanatory capability.

The principal limitation is the absence of experimental validation. Future evaluation should compare alternative lightweight models using representative IIoT traffic, measure inference latency and resource utilization at the edge, quantify explanation overhead, and assess false-positive behavior under changing industrial workloads. Such testing is necessary to determine whether the conceptual advantages translate into measurable operational improvements.

CONCLUSION

This paper presented a lightweight explainable AI framework for real-time threat detection in Industrial Internet of Things networks. The proposed architecture integrates distributed data processing, lightweight feature engineering, hierarchical machine-learning detection, explainable outputs, edge-cloud workload allocation, and adaptive response coordination.

The central contribution is the treatment of cybersecurity detection as a multi-objective systems problem rather than a classification problem alone. The analysis demonstrates that distributed processing can support scalability, edge-based screening can improve responsiveness, cloud resources can support intensive analytical workloads, and explainability can improve the operational usefulness of machine-generated alerts. The supplied literature concerning cloud computing, serverless architectures, machine-

learning services, infrastructure automation, and scalable analytics provides architectural foundations for these principles.

The framework also identifies important trade-offs. Serverless execution can provide elasticity but may not be appropriate for latency-critical inference. Cloud analytics can provide computational capacity but can increase communication and operational costs. Explainability can improve trust but may introduce additional computation. Kubernetes and automated infrastructure management can improve scalability but may increase deployment complexity. Therefore, an effective IIoT security architecture must allocate computational responsibilities according to operational requirements rather than adopting a single technology universally.

The work is conceptually aligned with the broader evolution toward hybrid intrusion-detection systems in Industry 5.0 environments (Govindarajan et al., 2026), while placing particular emphasis on lightweight execution and explainability. Future research should implement the framework using representative IIoT datasets and real or emulated industrial environments, evaluate multiple lightweight detection algorithms, measure edge resource consumption, quantify detection latency and false-positive rates, and investigate explanation fidelity under changing operational conditions. Such empirical validation would provide the basis for transforming the proposed conceptual architecture into a deployable security framework.

REFERENCES

1. M. R. Bell, “Big Data Analytics Using Amazon Web Services (AWS),” International Journal of Cloud Computing and Services Science, 2013.

2. R. Buyya, C. S. Yeo, and S. Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities," 10th IEEE/ACM International Conference on Cluster, Grid and Cloud Computing, 2010.

3. M. E. Chowdhury and R. E. B. Hill, “Serverless Computing: Economic and Architectural Impact,” IEEE International Conference on Cloud Computing, 2016.

4. J.Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," OSDI'04: 6th Symposium on Operating System Design and Implementation, 2004.

5. Donaldson, D. L., Krishnan, R., Narayana, S., et al. "Serverless Computing: Economic and Architectural Impact." ACM Computing Surveys, 2021.

6. Kapoor, A. R., Kumar, V. S. "Automating Infrastructure Management with AWS CloudFormation." Journal of Cloud Infrastructure Management, 2023.

7. Kapoor, A., Sharma, S., et al. "Machine Learning with Amazon SageMaker: A Comprehensive Review." IEEE Access, 2022.

8. Moolchandani, S. H., Gupta, T. R. "Leveraging AWS Lambda for Serverless Data Processing Pipelines." Cloud Computing Journal, 2022.

9. Rao, N. S., Patel, M. L. "Real-Time Big Data Analytics using AWS EMR and Apache Spark." International Journal of Data Science and Analytics, 2023.

10. Reddy, P. V. S., Bhat, S. S. "A Comprehensive Survey on Machine Learning Solutions Powered by AWS." International Journal of Machine Learning and Computing, 2023.

11. Sharma, N., Gupta, M. K. "Advancements in Serverless Computing with AWS Lambda." Journal of Cloud Journal of Artificial Intelligence and Cyber Security (JAICS) An International Open Access, Peer-Reviewed, Refereed Journal

Volume No.6, Issue No.1 (2025) 9 Computing and Services Science, 2023.

12. Singh, H., Choudhury, P., Joshi, V. "AWS and Kubernetes: A Cloud-Native Approach for Scalable, Resilient Systems." Journal of Cloud Computing, 2021.

13. Smith, P. B., Doe, J. K., Rogers, R. H. "Optimizing Cloud Cost Management in AWS: A Review and Future Directions." IEEE Transactions on Cloud Computing, 2023.

14. J. Xu, Y. Chen, and Z. Zhao, "A Comprehensive Study of Amazon Web Services (AWS) Adoption," Journal of Cloud Computing, 2017.

15. S. J. Yau and P. M. Chen, "Cost-Aware Cloud Computing for Elastic Services," IEEE Transactions on Cloud Computing, 2016.

16. Govindarajan, V., Ahmed, F., Faheem, Z. B., Bilal, M., Ayadi, M., & Ali, J. (2026). Aegis-5: A Hybrid Ensemble Framework for Intrusion Detection in Industry 5.0 Driven Smart Manufacturing Environment. ACM Transactions on Autonomous and Adaptive Systems.

17. K. Ramamurthy, N. Bellamkonda and N. Amanmadov, "ScalePulse: Combinatorial Llm Framework for Scalability Constraint," SoutheastCon 2026, Huntsville, AL, USA, 2026, pp. 1-6, doi: 10.1109/SoutheastCon63549.2026.11476075

18. Philip, P. G. (2025). Explainable Artificial Intelligence (XAI) for Project Governance: Improving Transparency and Stakeholder Trust in Automated Project Decision. Journal of Project Management Studies, 2(1), 37–55. https://doi.org/10.58425/jpms.v2i1.570.