A Unified Deep Temporal Inference and Rule-Based Decision Framework for Resource-Aware Task Scheduling in Edge–Fog Networks

B. Lakshman Rao[1], P. Wajid[2], Sk. Umar Farooq [2], Sk. Vasi UR Rahman[2], Sk. Abubakar Siddiq[2], Yeneti Lokesh[2]

[1]Professor, [2]UG Student, [1,2]Department of Computer Science and Engineering (AI-ML)

[1,2]Geethanjali Institute of Science and Technology, Nellore-Bombay Highway, S.P.S.R, Andhra Pradesh 524137, India

To Cite this Article

B. Lakshman Rao, P. Wajid, Sk. Umar Farooq, Sk. Vasi UR Rahman, Sk. Abubakar Siddiq, Yeneti Lokesh, "A Unified Deep Temporal Inference and Rule-Based Decision Framework for Resource-Aware Task Scheduling in Edge–Fog Networks", Journal of Science Engineering Technology and Management Science, Vol. 03, Issue 04, April 2026,pp: 1000-1014, DOI: http://doi.org/10.64771/jsetms.2026.v03.i04.pp1000-1014

Submitted: 08-03-2026 Accepted: 16-04-2026 Published: 23-04-2026

ABSTRACT

The Internet of Things (IoT) has transformed into a vast and dynamic ecosystem in which billions of interconnected devices continuously exchange data to enable real-time monitoring, intelligent decision-making, and automated control. These devices operate through Local Area Networks (LAN), Internet-based communication, and Ad-hoc networks, generating massive volumes of sensor data that demand efficient processing to maintain system responsiveness. Conventional cloud-centric approaches face limitations such as high latency, bandwidth constraints, and increased energy consumption, making them unsuitable for time-sensitive applications including healthcare, autonomous systems, and industrial automation. To address these challenges, a hierarchical computing paradigm integrating Edge, Fog, and Cloud layers has been adopted. Edge computing ensures low-latency processing near data sources, while fog computing provides intermediate computational support for data aggregation and analysis, and the cloud enables large-scale analytics and long-term storage. However, optimal task scheduling across these layers remains a critical challenge. Existing approaches such as Passive Aggressive Classifier (PAC), Naive Bayes Classifier (NBC), and K-Nearest Neighbors (KNN) provide baseline solutions but lack adaptability in dynamic environments. To overcome these limitations, a hybrid Convolutional Recurrent Network with Greedy Rule Interpretable Machine (CRN-GRIM) is proposed, combining deep spatio-temporal learning with interpretable rule-based decision-making. This approach enhances task allocation efficiency by considering priority, resource demand, and network conditions, thereby improving system scalability, latency reduction, and overall performance in IoT-driven Edge–Fog ecosystems.

Key words: Task scheduling. Edge-fog environments, Temporal learning, Local area networks, CRN-GRIM, Hierarchical computing.

1. INTRODUCTION

The Internet of Things (IoT) has evolved into a vast technological ecosystem in which interconnected devices communicate continuously to support real-time operations, intelligent decision-making, and automated responses. According to recent research on multi-layer IoT architecture, modern devices rely on communication models such as LAN, Internet-based communication, and Ad-hoc networking to exchange data seamlessly across distributed environments. IoT devices generate large volumes of raw data, including sensor readings, status updates, and environmental changes, all of which must be processed efficiently to maintain smooth system operation.

Because traditional cloud-centric processing introduces latency and increases bandwidth consumption, researchers have proposed a hierarchical computing architecture consisting of Edge, Fog, and Cloud layers. These layers collectively manage data collection, filtering, storage, and analysis, ensuring that tasks are assigned to the most suitable computing level based on urgency, complexity, and resource availability. Edge computing performs immediate processing near the data source, reducing transmission delay and enabling real-time responsiveness. Fog computing serves as an intermediary layer that aggregates, refines, and performs medium-level analysis on incoming data streams, helping offload tasks from both edge devices and cloud servers. Cloud computing, as the final tier, provides large-scale storage and heavy computational capability needed for complex analytics, long-term archiving, and integration across multiple IoT domains. These three layers create a continuous flow of information that supports efficient data sharing, reduces system bottlenecks, enhances reliability, and improves the overall performance of IoT networks.

In Figure 1.1 illustrate this three-tier architecture, edge devices initiate the data lifecycle by generating raw information through sensors and embedded components. Studies show that edge computing significantly benefits delay-sensitive applications by enabling quick, localized decisions and preventing unnecessary communication with distant cloud servers. However, edge devices often suffer from resource limitations, making them unsuitable for tasks requiring advanced analytics or large memory consumption. As a result, fog computing bridges this gap by offering more processing power closer to the network edge. Fog nodes can temporarily store data, perform aggregation, and apply rule-based or machine-learning algorithms to determine whether data should remain local or be forwarded to the cloud. Through this distributed model, fog computing reduces backbone network traffic and ensures that only meaningful information reaches the cloud. Research further demonstrates that fog computing improves the efficiency of IoT systems by enhancing task distribution, supporting location-awareness, and enabling parallel processing across nodes. Once refined data reaches the cloud, it undergoes large-scale processing, batch computation, machine-learning training, and long-term preservation. Cloud platforms integrate data from multiple fog domains, allowing cross-application collaboration, global analytics, and historical insights. This layered structure ensures that IoT systems maintain a balance between speed, accuracy, resource usage, and scalability while ensuring that each component performs tasks suited to its capabilities. The entire process enhances the reliability of IoT operations, making them more adaptable to dynamic conditions.

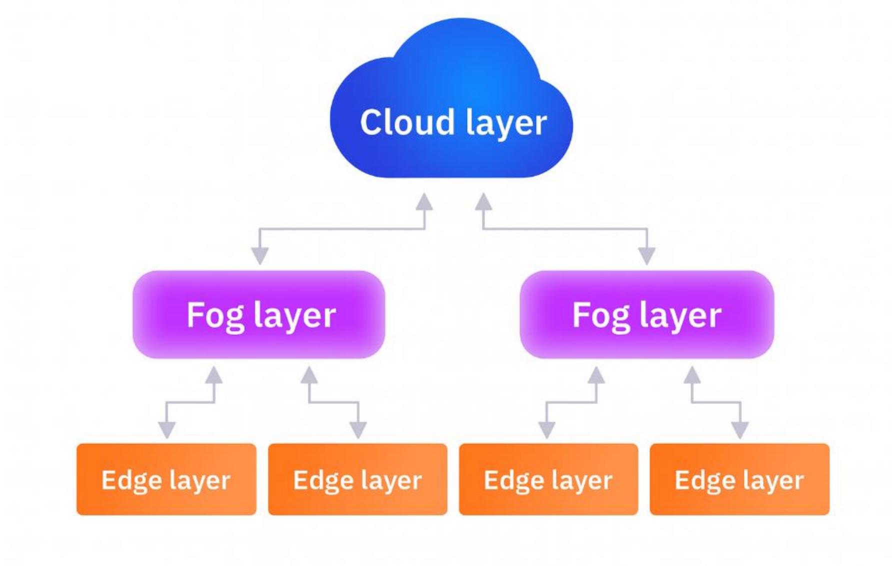

Figure 1. Edge-Fog Layer Architecture

As emphasized across open-access research literature, efficient task allocation remains one of the most critical challenges in IoT architecture. When tasks are not assigned properly across edge, fog, and cloud layers, systems face increased delays, unnecessary energy consumption, bandwidth overload, and inconsistent performance. Achieving optimal task allocation ensures that each computing layer handles the operations it is best suited for: edge devices respond instantly to time-critical events, fog nodes handle intermediate processing and coordination, and cloud servers undertake deep analytics and long-term storage. This collaborative model supports smart decision-making by ensuring the continuous movement of data between layers—flowing upward from devices to fog to cloud for analysis, and downward from cloud to fog to devices in the form of decisions or control commands. Such a bi-directional data-sharing approach strengthens IoT ecosystems by enabling adaptive, context-aware operations. Researchers conclude that the success of IoT deployments depends on how effectively data is shared, processed, and managed across these hierarchical layers. When combined, Edge, Fog, and Cloud computing form a unified continuum that improves throughput, reduces latency, enhances scalability, and ensures that IoT networks operate with maximum efficiency. This integrated architecture allows applications in healthcare, smart cities, transportation, industry, agriculture, and environmental monitoring to function reliably even under massive data loads, ultimately demonstrating why the three-layer data-sharing framework is considered the backbone of modern IoT systems.

2. LITERATURE SURVEY

Rajkumar Buyya et al. [1] The Internet of Everything paradigm is being rapidly adopted in developing applications for different domains like smart agriculture, smart city, big data streaming, and so on. These IoE applications are leveraging cloud computing resources for execution. Fog computing, which emerged as an extension of cloud computing, supports mobility, heterogeneity, geographical distribution, context awareness, and services such as storage, processing, networking, and analytics on nearby fog nodes. Adhikari M. et al. [2] the fog server in a fog computing paradigm extends cloud services to latency-sensitive tasks by employing fog nodes (FNs) near user devices. The resource-constrained FNs face the challenge of meeting stringent deadlines of latency-sensitive tasks. The completion deadline of such tasks becomes critical on pre-emption. Task preemption is unavoidable in uncertain events, such as FN hostility, overloading, and mobility of the host FN or the user device. Rescheduling the task that is likely to face preemption is a better solution than terminating it. Qadri YA et al. [3] The history of human development has proven that medical and healthcare applications for humanity always are the main driving force behind the development of science and technology. The advent of Cloud technology for the first time allows providing systems infrastructure as a service, platform as a service and software as a service. Cloud technology has dominated healthcare information systems for decades now. However, one limitation of cloud-based applications is the high service response time.

Chidamber et al. [4] Cloud computing is a highly popular computing technique. Cloud combined with IoT, fog, edge, and mist computing in 5G networks gives us realtime and highly predictive responses leading to a better and smart life. It requires a highly robust and integrated cloud administration, especially cloud resource allocation. Artificial intelligence and machine learning can be easily implemented along cloud design patterns for efficient resource allocation. In this paper we discuss multi-tenant cloud resource allocation problem. Sethi P et al. [5] The Internet of Things (IoT) applications and services are increasingly becoming a part of daily life; from smart homes to smart cities, industry, agriculture, it is penetrating practically in every domain. Data collected over the IoT applications, mostly through the sensors connected over the devices, and with the increasing demand, it is not possible to process all the data on the devices itself. The data collected by the device sensors are in vast amount and require high-speed computation and processing, which demand advanced resources.

Mohamed Abd Elaziz et al. [6] Fog computing, as a distributed paradigm, offers cloud-like services at the edge of the network with low latency and high-access bandwidth to support a diverse range of IoT application scenarios. To fully utilize the potential of this computing paradigm, scalable, adaptive, and accurate scheduling mechanisms and algorithms are required to efficiently capture the dynamics and requirements of users, IoT applications, environmental properties, and optimization targets.

David Berend et al. [7] This tutorial presents a performance engineering approach for optimizing the Quality of Service (QoS) of Edge/Fog/Cloud Computing environments using AI and Coupled-Simulation being developed as part of the Co-Simulation based Container Orchestration (COSCO) framework. It introduces fundamental AI and co-simulation concepts, their importance in QoS optimization and performance engineering challenges in the context of Fog computing. Hong, C.-H. et al [8] Artificial Intelligence (AI) is redefining resource management across edge, fog, and cloud computing systems by enabling dynamic, predictive, and autonomous decision-making. This paper explores emerging AI-augmented strategies designed to optimize latency, energy consumption, workload distribution, and quality of service (QoS). Traditional heuristic-based algorithms, while foundational, often fall short in handling heterogeneous, dynamic environments characterized by variable loads and tight latency constraints. AI models—ranging from Support Vector Machines (SVM) and reinforcement learning (RL) to clustering and regression techniques—have demonstrated superior adaptability through workload prediction, anomaly detection, and optimized resource provisioning. M. T. Mardini et al. [9] The Internet of Things (IoT) is revolutionizing numerous industries, including healthcare services, known as the Internet of Medical Things (IoMT). A large amount of generated data in IoMT applications need to be transmitted, analysed, and stored. Consequently, the cloud-only architecture was proposed as being the best-fit organizational infrastructure. Indeed, cloud capabilities of processing, networking, and storage are overwhelming properties that make it outperform classical solutions for decades when it comes to healthcare applications. Nevertheless, this architecture could not keep up with the ever-growing amount of biomedical data. One of the main drawbacks of cloud architecture is the large latency, which prevents it from delivering real-time alerts to save the patient's life in critical situations.

Mahmood Z et al. [10] In recent years, fog computing has emerged as a computing paradigm to support the computationally intensive and latency-critical applications for resource limited Internet of Things (IoT) devices. The main feature of fog computing is to push computation, networking, and storage facilities closer to the network edge. L. Niu, X ET AL-[11] nhis dynamic environment demands efficient scheduling mechanisms that can adapt to user movement while meeting application deadlines and optimizing edge resource utilization. This paper proposes an approach for scheduling based on Deep Reinforcement Learning, specifically using an Advantage Actor-Critic architecture within a Fog and Edge computing framework for IoT applications. The method enables distributed decision-making by deploying actor agents at edge nodes and a centralized critic at the fog node, facilitating continuous adaptation through system-wide feedback. S. A. Celtek and A. Durdu et al. [12] the integration of software-defined networking (SDN) and cloud radio access networks (CRANs) into vehicular ad hoc networks (VANETs) presents intricate challenges to achieving stringent service level objectives (SLOs). These objectives include optimizing data flow and resource management, achieving low latency and rapid response times, and ensuring network resilience under fluctuating conditions. Traditional load balancing and clustering approaches, designed for more static environments, fall short in the dynamic and variable context of VANETs.

T.-Y. Kim et al .[13] The advent of wearable sensor technology, like oximeters, accelerometers, and gyroscope-based monitoring devices, are being widely used to address the important problem of health monitoring for mobility patients. Unusual patient data prompts healthcare facilities to take immediate action by sending out alerts. The major goal of this paper is to investigate the field of fog computing-assisted healthcare service with regard to priority-aware healthcare task offloading. When it comes to providing end users with extra processing power for time-sensitive tasks, the fog server is essential. But this ever-changing environment is really hard when some jobs demand a far slower response time than others. This problem is solved by introducing a priority-aware scheduling and task offloading technique that prioritizes high-priority jobs, especially those with strict deadlines, when allocating CPU resources. Badidi E et al [14]The significance of this research comes in the fact that it has the ability to completely change the healthcare system by relocating computing resources closer to the data source, hence facilitating more rapid and accurate analysis of medical data. Latency, privacy concerns, and inability to scale are common in traditional cloud-centric techniques. Feng, C et al [15] The management of decentralized energy resources and smart grids needs novel data-driven low-latency applications and services to improve resilience and responsiveness and ensure closer to real-time control. However, the large-scale integration of Internet of Things (IoT) devices has led to the generation of significant amounts of data at the edge of the grid, posing challenges for the traditional cloud-based smart-grid architectures to meet the stringent latency and response time requirements of emerging applications. In this paper, we delve into the energy grid and computational distribution architectures, including edge–fog–cloud models, computational orchestration, and smart-grid frameworks to support the design and offloading of grid applications across the computational continuum.

3. PROPOSED SYSTEM

The system architecture presents the overall structural design and workflow of the proposed intelligent task scheduling framework for edge–fog computing environments. It explains how task-related data is collected, processed, analysed, and transformed into scheduling decisions using machine learning and deep learning models. The architecture follows a step-by-step approach starting from dataset preparation and feature analysis to model training, performance evaluation, and real-time deployment through a web-based application. This structured architecture ensures efficient data flow, accurate task classification, reduced latency, and improved resource utilization across edge, fog, and cloud layers.

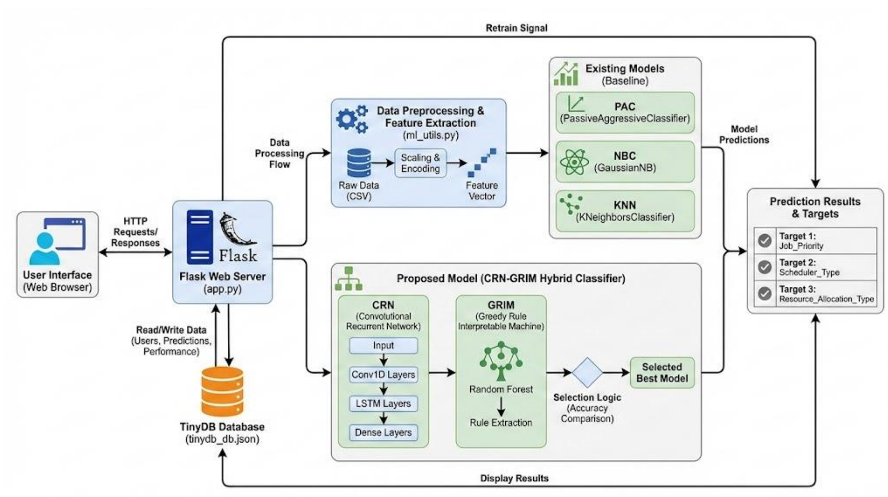

Fig 2. Proposed system architecture of Task scheduling

The system begins with collecting IoT- and network-based task scheduling data containing attributes like task priority, execution time, scheduling type, and resource requirements. The collected data is then preprocessed through missing value handling, noise removal, encoding, and normalization to ensure quality and consistency. Exploratory Data Analysis (EDA) is performed to identify patterns, correlations, and important features for model development. The processed data is initially evaluated using existing classifiers such as Passive Aggressive Classifier (PAC), Naive Bayes Classifier (NBC), and K-Nearest Neighbors (KNN) to establish baseline performance. The proposed CRN-GRIM model, which combines Convolutional Recurrent Networks for feature extraction and Greedy Rule Interpretable Machine for transparent decision-making, is then applied to enhance scheduling accuracy and interpretability. The performance of all models is compared using key metrics, and the best-performing model is selected for prediction on unseen test data to ensure generalization. Finally, the system is deployed using a Flask-based web application, enabling real-time task scheduling predictions through an interactive user interface.

3.2 Hybrid CNN-GRIM model

3.2.1 Convolutional Recurrent Network Features

The Convolutional Recurrent Network (CRN) is integrated into the proposed deep decision framework to enable intelligent task scheduling in Edge–Fog architectures by jointly learning spatial and temporal task characteristics. Incoming tasks are represented as time-series feature matrices that capture variations in execution time, workload intensity, network latency, and resource utilization over time. The convolutional layers perform local feature extraction by identifying meaningful patterns and correlations among task attributes, while pooling layers reduce dimensionality and improve computational efficiency. This hierarchical feature learning process allows the CRN to generate compact and informative representations suitable for dynamic scheduling environments.

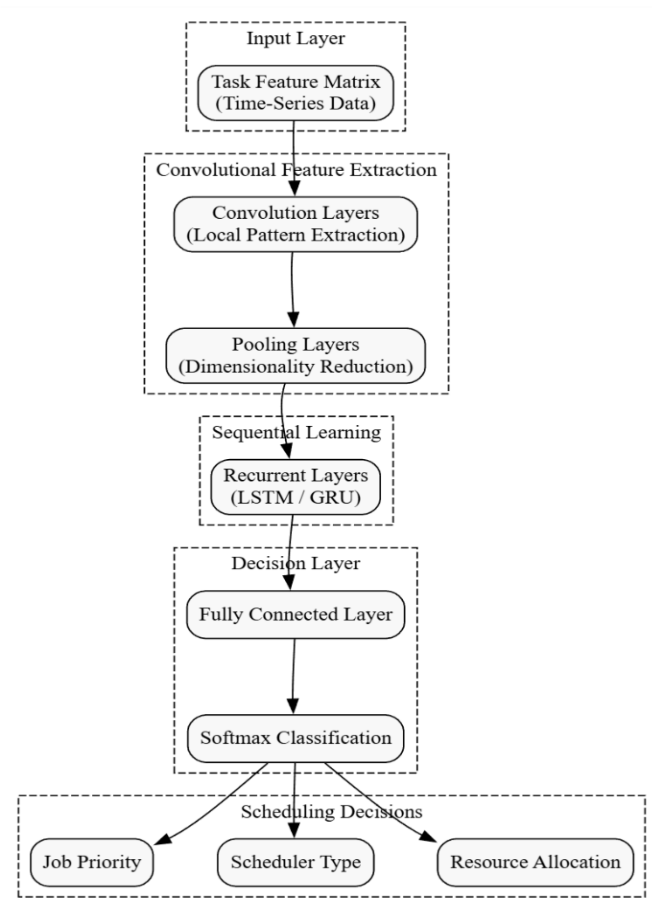

Fig 3. Internal workflow of CRN

The CRN process begins with an input layer where tasks are represented as a time-series feature matrix capturing both static and dynamic attributes such as execution time, workload variation, latency, and resource utilization over time. Convolutional layers are then applied to extract spatial patterns and correlations among task features, followed by pooling to reduce dimensionality while preserving important information. These extracted features are passed to recurrent layers like LSTM or GRU, which learn temporal dependencies and sequential workload patterns in Edge–Fog environments. Finally, the learned representations are processed through fully connected layers with a SoftMax function to generate probability scores, enabling the system to make accurate scheduling decisions such as assigning job priority and selecting the appropriate scheduler type.

3.3. Greedy Rule Interpretable Machine

The Greedy Rule Interpretable Machine (GRIM) is incorporated into the proposed deep decision framework to address the need for explainable and transparent task scheduling in Edge–Fog architectures. In highly dynamic and distributed environments, scheduling decisions directly impact latency, energy consumption, and system reliability, making interpretability a critical requirement. GRIM adopts a greedy rule-learning strategy that incrementally constructs decision rules by selecting task features that provide the highest information gain at each stage of learning. This process enables the model to capture dominant scheduling patterns using a compact set of human-readable rules rather than complex numerical representations. Task attributes such as execution time, deadline constraints, data size, communication delay, bandwidth availability, and node resource status are systematically evaluated to form concise decision conditions. As a result, GRIM produces transparent rule sets that clearly describe how specific task characteristics influence scheduling outcomes, allowing stakeholders to easily understand, validate, and audit the decision logic employed by the framework.

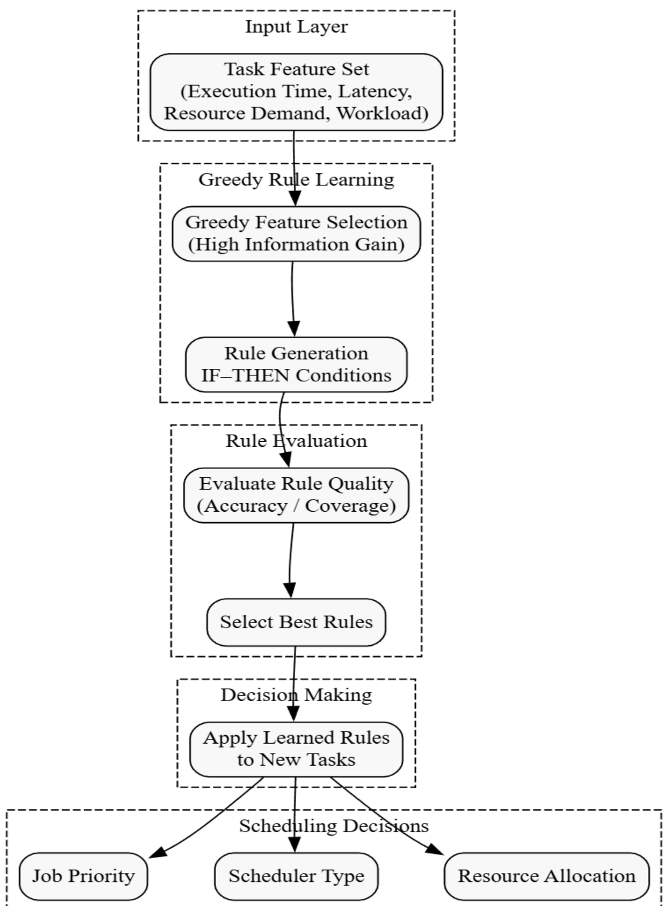

Figure 4. Internal workflow of GRIM

The process begins with the input layer, where task features such as execution time, latency, resource demand, and workload describe each task's requirements. A greedy feature selection approach is then applied to choose the most informative attributes based on information gain, reducing complexity while retaining critical data. Using these selected features, the system generates interpretable IF–THEN rules that capture task behavior patterns. Each rule is evaluated using metrics like accuracy and coverage to assess its effectiveness, and only the best-performing rules are retained. These optimized rules are then applied to incoming tasks, enabling the system to make intelligent scheduling decisions. Finally, based on rule matching, the system determines job priority, selects the appropriate scheduler (edge or fog), and allocates optimal resources for efficient task execution.

4. RESULT ANALYSIS

Figure 5 shows the Data Source Distribution pie chart illustrates the proportional contribution of different task origins within the Edge–Fog workload dataset, highlighting the heterogeneous nature of incoming tasks. The distribution is nearly uniform across all four data sources, indicating a balanced workload scenario. Enterprise DB tasks constitute the largest share at approximately 25.9%, reflecting consistent enterprise-level data processing demands. IoT sources closely follow with about 25.6%, emphasizing the significant role of sensor-driven and real-time data in Edge–Fog environments. Social Media tasks account for around 24.4%, representing user-generated and interaction-driven workloads that often exhibit variable traffic patterns.Cloud-originated tasks contribute roughly 24.1%, indicating background and centralized service requests.

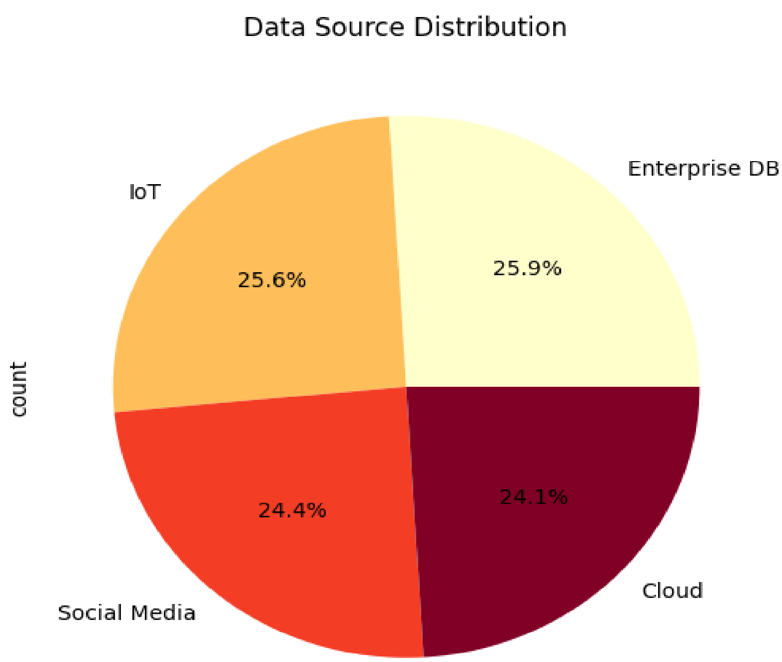

Figure 5. Data Source Distribution

Figure 6 shows the correlation matrix shows that most task and system features have very weak correlations with each other, as indicated by off-diagonal values close to zero, while diagonal values of 1.00 represent perfect self-correlation. This indicates that features such as error rate, CPU utilization, memory usage, execution time, throughput, waiting time, active users, network bandwidth, and data source are largely independent. Only minor positive relationships appear between execution time and memory consumption, and between network bandwidth utilization and active users, reflecting realistic workload behaviour. Overall, the low correlation confirms minimal feature redundancy, which improves the effectiveness and stability of machine learning–based task scheduling in the Edge–Fog architecture

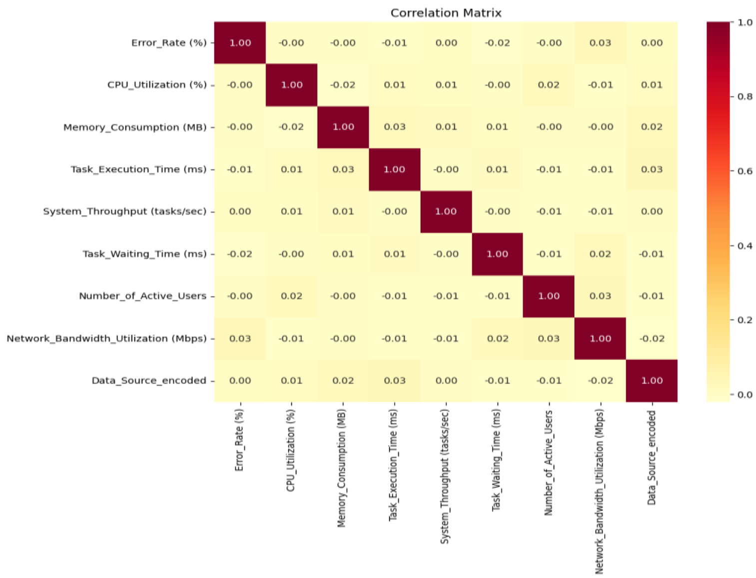

Figure 6. Correlation heatmap

Figure 7. present the confusion matrices of the CRN-GRIM classifier for three scheduling classes: High, Low, and Medium, highlighting their prediction accuracy and misclassification patterns. The CRN-GRIM model demonstrates superior performance with highly concentrated diagonal values of 920, 913, and 991 for High, Low, and Medium respectively, and minimal misclassification, confirming its effectiveness for intelligent task scheduling.

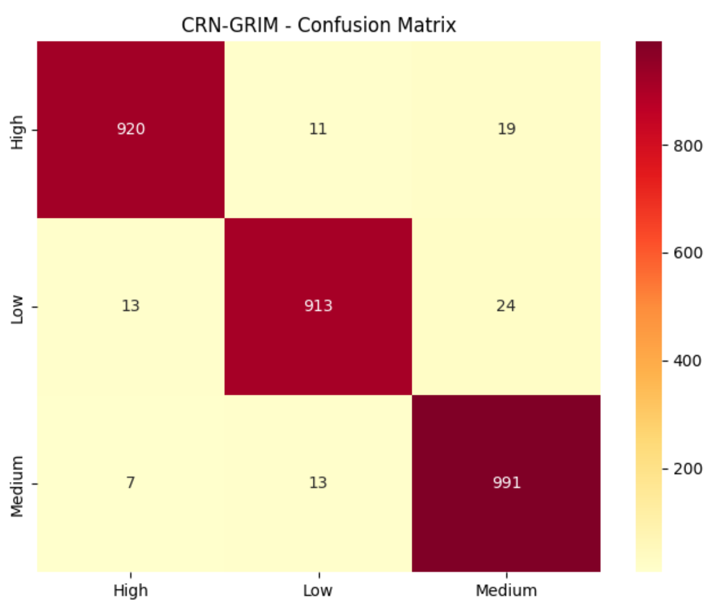

Figure 7: Confusion matrix obtained using proposed CNN-GRIM model

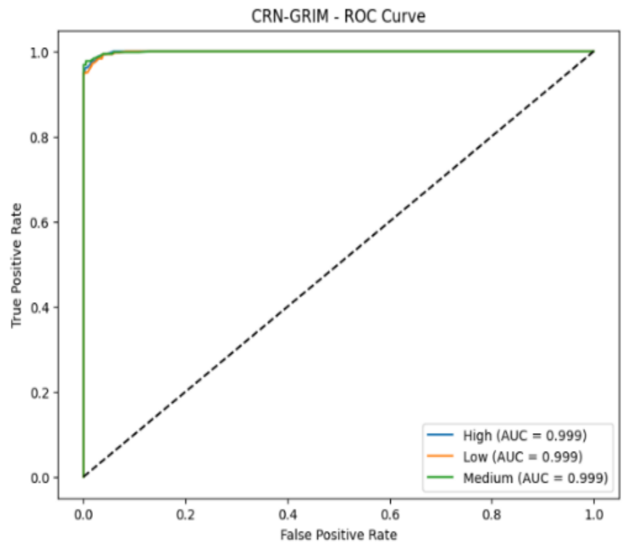

Figure 8. ROC curve obtained using proposed CNN-GRIM model

Figure 9. illustrates the confusion matrix analysis of CNN-GRIM model used for scheduler type prediction across four scheduling classes: ASB-Dynamic-CapsNet, FCFS, Priority-Based, and Round Robin. CNN-GRIM significantly outperforms all other methods by achieving very high diagonal values for all scheduling classes while maintaining minimal misclassification. This indicates strong learning capability, accurate class discrimination, and robustness under dynamic task conditions. Overall, Figure 9.6 clearly confirms that CRN-GRIM is the most effective model for intelligent scheduler selection when compared with PAC, NBC, and KNN classifiers.

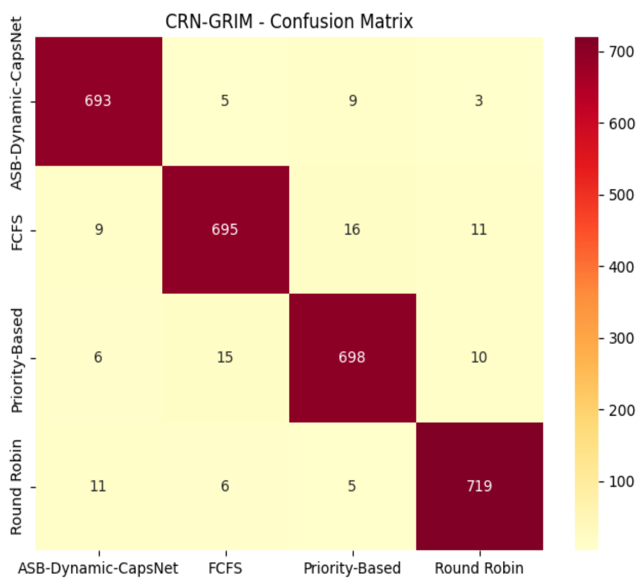

Figure 9. Confusion Matrix of Scheduler Type Classification using Proposed CRN-GRIM

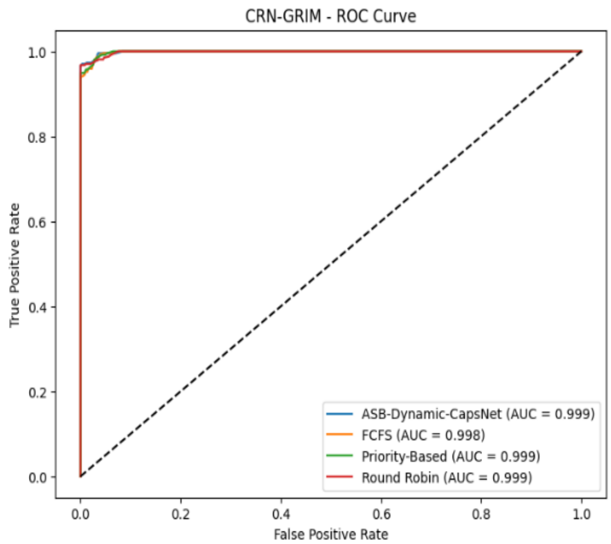

Figure 10. ROC Curves of Scheduler Type Classification Output using Proposed CRN-GRIM

The figure 11 illustrate the performance of CRN-GRIM classifier for Dynamic and Static scheduling classification using confusion matrices and ROC curves. The CRN-GRIM model achieves outstanding performance with highly concentrated diagonal values of 1394 (Dynamic–Dynamic) and 1434 (Static–Static), minimal misclassification (only 52 and 31 errors), and an almost perfect ROC curve with an AUC of 0.999, clearly demonstrating its superior accuracy, robustness, and reliability for scheduling type prediction.

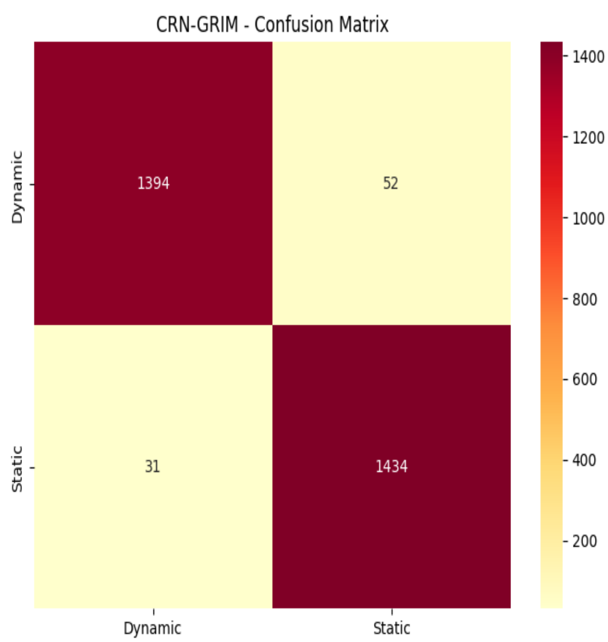

Figure 11. Confusion matrix obtained for resource allocation using CNN-GRIM model

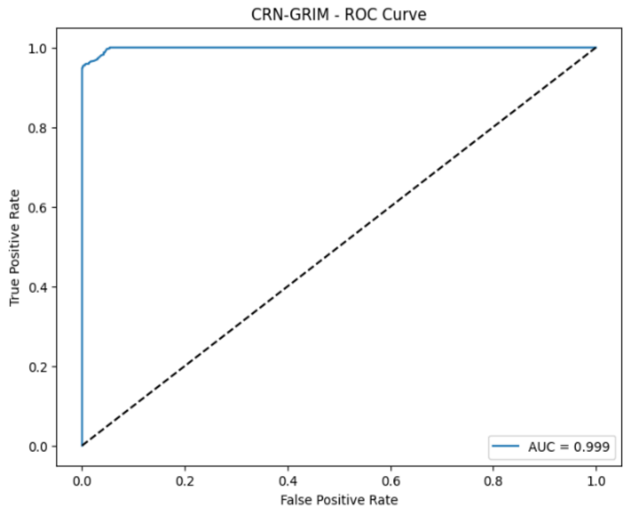

Figure 12. ROC curve obtained using CNN-GRIM model

Figure 13 shows the test data input interface of the proposed intelligent task scheduling system, where specific sample values are entered to evaluate the prediction mechanism. The input parameters include Error Rate = 10%, CPU Utilization = 10%, Memory Consumption = 10 MB, Task Execution Time = 10 ms, System Throughput = 10 tasks/sec, Task Waiting Time = 10 ms, Number of Active Users = 10, Network Bandwidth = 10 Mbps, and Data Source = Cloud. These controlled and uniform input values enable consistent testing of the scheduling models under a low-load and stable system condition.

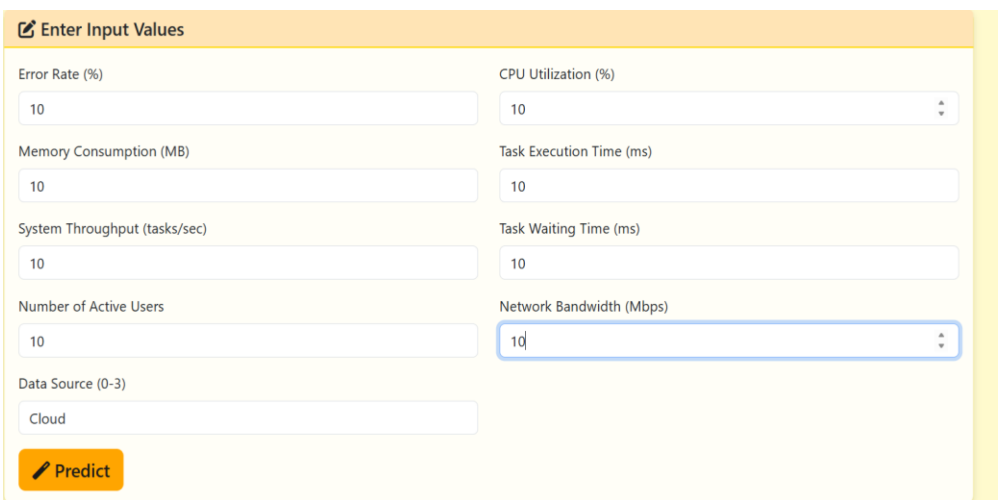

Figure 13. Test data input interface

Figure 14 shows the prediction results generated for the given test inputs by different classifiers integrated into the framework. For Job Priority, the PAC classifier predicts High, whereas NBC, KNN, and CRN-GRIM classify the task as Medium priority. In terms of Scheduler Type, PAC and NBC recommend Priority-Based scheduling, while KNN and CRN-GRIM select Round Robin scheduling. For Resource Allocation Type, PAC and NBC predict Dynamic allocation, whereas KNN and CRN-GRIM choose Static allocation. This comparison highlights the variation in decision-making among traditional classifiers and the more consistent predictions produced by advanced models under identical input conditions

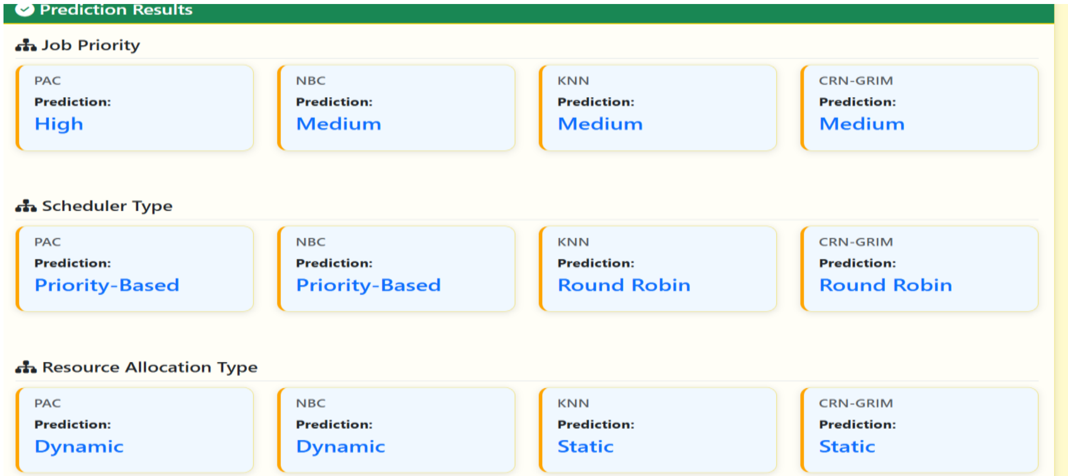

Figure 14. Prediction Result obtained for all the models

5. CONCLUSION

The proposed intelligent task scheduling framework based on a Deep-Decision Fusion approach has been successfully designed, implemented, and evaluated in an Edge–Fog computing environment. The experimental analysis was conducted using a dataset of 14,553 task records and multiple system parameters such as CPU utilization (average 49.72%), memory consumption (mean 4217 MB), task execution time (mean 2506 ms), and system throughput (mean 5.27 tasks/sec). Comparative evaluation among traditional classifiers (PAC, NBC, and KNN) and the proposed CRN-GRIM model clearly demonstrates the superiority of the hybrid deep learning approach. While PAC and NBC achieved low accuracies in the range of 25–35%, and KNN showed moderate performance with accuracies around 48–62%, the CRN-GRIM model consistently achieved outstanding performance with accuracies of 97.01% for Job Priority, 96.36% for Scheduler Type, and 97.15% for Resource Allocation Type, along with equally high precision, recall, and F1-scores. Furthermore, confusion matrix and batch prediction analyses confirmed that CRN-GRIM produces highly reliable and stable decisions with minimal misclassification across High, Medium, and Low classes. The system successfully integrates real-time prediction, exploratory data analysis, and batch-level scheduling decisions through a user-friendly web interface. The results validate that deep feature extraction using convolutional and recurrent learning, combined with rule-based interpretability, significantly enhances scheduling accuracy and consistency. Overall, the proposed framework effectively reduces scheduling uncertainty, improves resource utilization, and supports intelligent decision-making in dynamic cloud and Edge–Fog computing environments.

REFERENCES

[1]. Rajkumar Buyya and Amir Vahid Dastjerdi. 2016. *Internet of Things: Principles and Paradigms*Elsevier.

[2]. Adhikari, M., Mukherjee, M., & Srirama, S.N. (2019). Dpto: A deadline and priority-aware task offloading in fog computing framework leveraging multi-level feedback queueing.

[3]. Qadri YA, Nauman A, Zikria YB, Vasilakos AV, Kim SW (2020) The future of healthcare internet of things: a survey of emerging technologies

[4]. Chidamber, S.R., Kemerer, C.F.: A metrics suite for object-oriented design. IEEE Trans. Software Eng. **20**(6), 476–493 (1994)

[5]. Sethi P, Sarangi SR (2017) Internet of things: architectures, protocols, and applications. J Electr Comput Eng 2017:9324035

[6]. Mohamed Abd Elaziz, Laith Abualigah, and Ibrahim Attiya. 2021. Advanced optimization technique for scheduling IoT tasks in cloud-fog computing environments

[7]. David Berend, Xiaofei Xie, Lei Ma, Lingjun Zhou, Yang Liu, Chi Xu, and Jianjun Zhao. 2020. Cats are not fish: Deep learning testing calls for out-of-distribution awareness. In Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering. 1041--1052.

[8]. Hong, C.-H., & Varghese, B. (2018). Resource Management in Fog/Edge Computing: A Survey. arXiv. arXiv

[9]. M. T. Mardini, Y. Iraqi, "A survey of healthcare monitoring systems for chronically ill patients and elderly," *Journal of Medical Systems*, vol. 43, no. 3, pp. 1–21, 2019

[10]. Mahmood Z, Ramachandran M (2018) Fog computing: concepts, principles and related paradigms. In: Mahmood Z (ed.) Fog computing: concepts, frameworks and technologies, chap. 1. Springer, Berlin, pp. 3–21

[11]. L. Niu, X. Chen, N. Zhang, Y. Zhu, R. Yin and C. Wu, "Multiagent Meta-Reinforcement Learning for Optimized Task Scheduling in Heterogeneous Edge Computing Systems," IEEE Internet of Things Journal, vol. 10, no. 12, pp. 10519-10531, 2023. doi.org/10.1109/JIOT.2023.3241222.

[12]. S. A. Celtek and A. Durdu, "A novel adaptive traffic signal control based on cloud/fog/edge computing," *Int. J. Intell. Transp. Syst. Res.*, vol. 20, no. 3, pp. 639–650, 2022.

[13]. T.-Y. Kim, S. Youm, J.-J. Jung, and E.-J. Kim, "Multi-hop wban construction for healthcare iot systems," in *2015 International conference on platform technology and service*. IEEE, 2015, pp. 27–28.

[14]. Badidi E, Mahrez Z, Sabir E (2020) Fog computing for smart cities' big data management and analytics: a review. Future Inter 12(11):190

[15]. Feng, C.; Wang, Y.; Chen, Q.; Ding, Y.; Strbac, G.; Kang, C. Smart grid encounters edge computing: Opportunities and applications. *Adv. Appl. Energy* **2021**, *1*, 100006.