**Bulletin of Electrical Engineering and Informatics**
Vol. 15, No. 2, April 2026, pp. 1313~1323
ISSN: 2302-9285, DOI: 10.11591/eei.v15i2.9885


# Multi objective energy aware integrated cloud scheduling with a consensus-based security

**Fairoz Pasha[1,2], Jayapandian Natarajan[1]**
[1]Department of Computer Science and Engineering, School of Engineering and Technology, Christ University, Kengeri Campus, Bangalore, India
[2]Department of Computer Science Engineering, School of Technology, Maulana Azad National Urdu University Polytechnic, Bangalore, India

**Article Info**

***Article history:***

Received Jan 13, 2025
Revised Jan 21, 2026
Accepted Feb 22, 2026

***Keywords:***

Cloud computing
Consensus-based security
Dynamic scheduling
Multi-objective
Workflow

**ABSTRACT**

This research presents a multi-objective, energy-aware workflow scheduling framework for heterogeneous cloud–edge environments that addresses both efficiency and data integrity challenges. Conventional encryption-based security mechanisms, although effective in protecting data during task offloading, often introduce significant computational and communication overhead, leading to degraded system performance. To overcome this limitation, this work proposes the consensus security-integrity and quality-aware workflow scheduler (CSIQA-WS), which integrates energy-aware scheduling with a lightweight, consensus-driven security mechanism. The model incorporates automatic service management and an attack prevention module to detect and mitigate malicious behavior during inter-node data transmission while maintaining quality of service (QoS) constraints. A dynamic coordination between edge and cloud resources enables efficient workload distribution and robust resource utilization. Experimental evaluation using scientific workflow benchmarks demonstrates that CSIQA-WS significantly reduces processing time and energy consumption compared to existing approaches. The proposed model achieves up to 92.29% reduction in processing time and consistently improves overall QoS while preserving data integrity in dynamic execution environments. These results indicate that CSIQA-WS provides an effective and scalable solution for secure and energy-efficient workflow scheduling in modern cloud–edge systems.



***Corresponding Author:***

Fairoz Pasha
Department of Computer Science and Engineering, School of Engineering and Technology
Christ University, Kengeri Campus
Bangalore, India
Email: fairoz.pasha@res.christuniversity.in

## 1. INTRODUCTION

The paradigm of computation as a service has evolved rapidly, with cloud computing emerging as a dominant model that enables users to store data remotely and access on-demand applications and services through the internet [1], [2]. By offering a shared pool of programmable computational resources [3], cloud computing delivers scalability, flexibility, and cost efficiency, allowing users to perform data storage, processing, and analytics without maintaining physical infrastructure [4], [5]. Virtualization plays a central role in this ecosystem by enabling multiple virtual machines (VMs) to share physical resources while maintaining isolation and service reliability [6].

Despite these advantages, the large-scale deployment of cloud services introduces critical challenges in resource allocation and quality of service (QoS) assurance [7], [8]. In modern data centers, numerous VMs compete for limited CPU, memory, and bandwidth resources hosted on physical machines (PMs), often leading to performance bottlenecks, underutilization, or overload conditions [8], [9]. To mitigate these issues, cloud providers employ intelligent scheduling, load balancing, and task offloading mechanisms to optimize resource utilization and meet QoS constraints [10]. However, aggressive task offloading and VM migration, while improving performance, expose cloud systems to significant security threats.

Security vulnerabilities such as distributed denial of service (DDoS) attacks, hypervisor exploits, data tampering, and live migration attacks pose serious risks during workflow execution and data transmission [11], [12]. Traditional security mechanisms, including encryption-based approaches, provide baseline protection but often incur high computational overhead, increased latency, and limited adaptability in dynamic cloud–edge environments [3], [13]. Consequently, emerging techniques based on machine learning, blockchain, and trust-aware security frameworks have gained attention for enabling real-time threat detection, integrity preservation, and decentralized security enforcement [14]–[16].

Another critical concern in cloud environments is energy efficiency. Large-scale data centers consume substantial energy, increasing operational costs and environmental impact. Energy-aware scheduling techniques, such as dynamic voltage and frequency scaling (DVFS), aim to reduce power consumption while maintaining performance [17]. However, existing DVFS-based models primarily focus on energy–performance trade-offs and often neglect security, trust, and integrity guarantees during workflow execution [18], [19]. Recent studies incorporating deep reinforcement learning and blockchain-based secure workflow scheduling (SWS) improve adaptability and traceability [20], [21] but suffer from high overhead, latency, or lack of convergence guarantees [22], [23] under adversarial conditions [24].

Traditional security mechanisms [25], [26], while effective to some extent, struggle to adapt to the scale and complexity of modern cloud environments [27]. Emerging solutions such as machine learning [28] and blockchain technologies have shown promise in addressing these challenges. Machine learning algorithms and Blockchain-trust-absed model namely SWS [29] can identify anomalous patterns and detect potential security breaches in real time, offering proactive defense mechanisms. Blockchain, with its decentralized and tamper-proof architecture, ensures secure data storage and transmission across distributed systems. Combined with trust models and consensus-based security protocols, these technologies provide robust frameworks to mitigate risks and enhance cloud security. The summary of exisitng method over propsoed CSIQA-WS is shown in Table 1.

Although prior works address energy efficiency, performance optimization, or security in isolation, there remains a clear gap in unified frameworks that jointly optimize energy consumption, QoS, and data integrity using lightweight, adaptive consensus mechanisms suitable for heterogeneous cloud–edge environments. Existing approaches lack scalable trust verification and robustness against dynamic attacks during multi-workflow execution.

This paper contributes a novel consensus security-integrity and quality-aware workflow scheduler (CSIQA-WS) that integrates multi-objective energy-aware scheduling with consensus-based security to ensure efficient, secure, and scalable workflow execution. By combining computation, communication, and offloading energy modeling with a noise-resilient consensus mechanism and attack-aware scheduling, CSIQA-WS outperforms DVFS-based and security-centric schedulers in both efficiency and integrity.

The significance of propsoed CSIQA-WS is as follows: the proposed CSIQA-WS framework provides practical guidance for designing secure, energy-efficient, and scalable workflow scheduling solutions in heterogeneous cloud–edge environments, supporting emerging scientific and data-intensive applications like Inspiral and CyberShake workflows under dynamic workloads and adversarial conditions. CSIQA-WS by efficiently handling task dependencies and QoS-aware scheduling, outperforming task-aware dynamic voltage and frequency scaling plus (DVFS-T+) [1], [25] and processor-aware dynamic voltage and frequency scaling plus (DVFS-P+) [13], [27] in real-time execution environments. The critical analysis showing that DVFS-T+ achieves energy savings of 15% but fails to maintain data integrity under 30% attack intensity. DVFS-P+ improves task completion time by 10% yet suffers up to 22% higher misclassification under adversarial conditions. These models lack adaptive consensus or trust-based verification, limiting their scalability in heterogeneous edge–cloud environments. By contrast, the proposed CSIQA-WS achieves over 95% detection accuracy and consistent QoS adherence, demonstrating superior robustness and multi-objective performance across dynamic workflow conditions.

By integrating communication, computation, and offloading energy into a multi-objective QoS function, CSIQA-WS reduces total energy consumption and operational cost across VMs compared to DVFS-based models, particularly in data-intensive tasks of CyberShake. The proposed CSIQA-WS achieves more precise task execution with fewer resource demands, thanks to its noise-resilient consensus mechanism,

enabling cost-efficient processing even under uncertain workloads, unlike DVFS-T+ and DVFS-P+, which lack convergence guarantees.

Table 1. Summary of existing methods vs. proposed CSIQA-WS

| Model | Focus | Strength | Limitation | Advancement in CSIQA-WS |
|---|---|---|---|---|
| DVFS-T+ [25] | Energy–deadline | Efficient power control | Lacks security and trust | Adds secure and adaptive scheduling |
| Blockchain model [26] | Workflow security | Strong traceability | High latency | Lightweight trust quantification |
| DVFS-P+ [27] | Energy–performance | Predictive control | Ignores integrity and QoS | Integrates quantitative security assurance |
| SWS [29] | Security–cost | Secure workflow mapping | High overhead | Lightweight and scalable security |
| DRL scheduler [28] | Energy–performance | Adaptive learning | No trust modeling | Multi-objective reward with security |
| CSIQA-WS (proposed) | Energy–security–scalability | Unified optimization | — | Secure, efficient, and adaptive scheduling |

This manuscript is organized as follows: section 1 presents the introduction and literature review on real-time scientific workflows. Section 2 details the proposed CSIQA-WS methodology. Section 3 discusses experimental results and performance comparisons. Finally, section 4 concludes the paper with key findings and future directions.

## 2. METHOD

If sufficient QoS is not provided by the respective PMs or servers, the task is offloaded to a new server, which may include an edge server. The connection between these servers can be either wired or wireless, which introduces security threats where malicious nodes may modify the data, thereby affecting the integrity of intermediate workflow data [25], [27]. To address these integrity issues, workflow data is generally encrypted and transmitted over the cloud platform during task offloading or execution. Yet, the strict QoS constraints enforced by workflow applications, combined with encryption-based security, lead to degraded performance due to increased data size and overhead in key generation and management. Recent methods [25], [27] have employed consensus-based security with promising results; nevertheless, these methods suffer from high misclassification rates, and limited work has been done on workflow execution in edge–cloud platforms. To address these research issues, this paper introduces a multi-objective energy-aware scheduling strategy integrated with consensus-based security for ensuring data integrity.

Figure 1 illustrates the consensus-based cloud security architecture. This architecture comprises multiple modules. Workload execution is managed through an automatic service manager module. A critical component is the attack manager module, which includes three subcomponents: attack monitor, attack detector, and attack preventer. The attack preventer functions as part of the attack manager module, executing real-time mitigation actions such as isolating malicious nodes, blocking compromised communication links, and redirecting workflows to trusted servers. The proposed module is integrated within the resource provisioning and scheduler components, which are directly connected to edge–cloud resources. This module acts as an interface between the workload–resource matchmaker and the cloud resource repository. The incoming data is generated using the resource generator, while the workload generator and the output are coordinated through the simulation initiator and controller, which are among the key components of this architecture. Together, these modules coordinate secure data transmission, workflow initialization, and attack mitigation, ensuring seamless operation, integrity preservation, and QoS-aware scheduling throughout the multi workflow execution.

In this section, we present the framework of a real-time workflow applications execution process. Consider a direct acyclic graph (DAG) $X$, in which there is various kinds of sets of tasks denoted by $K$, and each task may have a subtask which will be having a dependency denoted by $D$. This can be represented using (1):

$$X(K, D) \tag{1}$$

The set of real time tasks $K$ can be represented (2):

$$K = \{K_1, K_2, \dots, K_n\} \tag{2}$$

Also, the dependency of the subtasks $D$ can be represented using (3):

$$D = \{(K_q, K_r) | K_q, K_r \in K\} \tag{3}$$

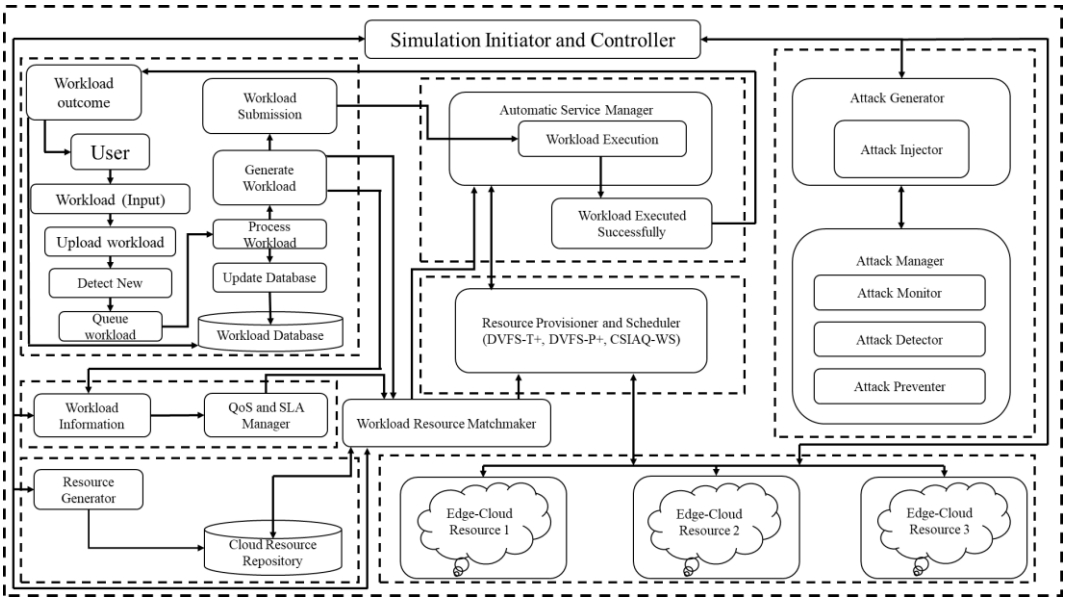


Figure 1. Proposed architecture model for execution of scientific workflow

The QoS of any task is decided based on the parameter which has been assigned to that task. Additionally, the QoS is specified for each task on its corresponding VM using accurate configuration that must be transmitted among tasks when taking into account each task's data-intensive workload. The data size that is interconnected between $K_q$ and $K_r$ is defined as $E(K_q, K_r)$, and the QoS aware executed or processing time of $K_q$ is defined as $S(K_q)$. Following is how the $K_q$ the task from earlier is accomplished which is shown using (4):

$$S(K_q) = \{K_q | (K_q, K_q) \in D\} \tag{4}$$

In (5) defines the incoming task $K_{\leftarrow}$ for the respective workload tasks.

$$F(K_{\leftarrow}) = \emptyset \tag{5}$$

and $K_{\rightarrow}$ represents the outgoing tasks using (6):

$$\nexists K_q \in K : K_{\rightarrow} \in F(K_q). \tag{6}$$

In this article a proposed multi-objective QoS metrics is defined that aims to minimize energy for computing on respective VMs, energy for communicating on respective VMs and energy for reconfiguring and offloading to new PM; the multi-objective QoS is defined through (7):

$$\gamma_l \triangleq \sum_{s=1}^{N} \gamma_{T_{\mathbb{C}}}(s) + \sum_{s=1}^{N} \gamma_{\mathbb{M}_{\mathbb{C}}}(s) + \sum_{s=1}^{N} \gamma_{F_{\mathbb{C}}}(s) \tag{7}$$

where $s$ defiens VM size, $N$ defines maximum number of VMs, $\gamma_{T_{\mathbb{C}}}(s)$ denotes the total energy consumed for workload processing and $\gamma_{\mathbb{M}_{\mathbb{C}}}(s)$ denotes the energy consumed for transferring workload data while taking the QoS constraint $I_t$ into account and $\gamma_{F_{\mathbb{C}}}(s)$ defines total energy consumed for offloading to new PMs $P$.

The proposed CSIQA-WS algorithm must meet convergence with precise average value; thus, noise must be decaying for assuring convergence. Further, for achieving better convergence outcome the asymptotic sum must be 0. The proposed security mechanism can work efficiently in unbiased manner for general noise and Gaussian noise using maximum likelihood estimator (MLE). The CSIQA doesn't prerequisite additional communication channel; thereby limiting resource usage and enhancing energy efficiency. The consensus mechanism ensures integrity and privacy as it requires few values to build consensus because computational node have more independence to optimize the noise $x(l)$. The assumption is described in following example, every computational node can select $x(l)$ jointly independently with exponential decaying covariance matrix.

In such cases, one has to guarantee that induced noise $x(l)$ doesn't impact consensus outcomes (that is., $x(l)$ should be correlated) for obtaining exact average consensus. The proposed CSIQA-WS model enables better scheduling by meeting security and quality trade-off constraints for executing both Inspiral and CyberShake workflows in a heterogeneous cloud environment more efficiently compared to other baseline methods such as DVFS-T+ [25] and DVFS-P+ [27].

Algorithm 1. Algorithmic flow for CSIQA-WS

1. **Initialize** system state, resource repository, and attack manager modules (monitor, detector, preventer).
2. **Preprocess DAG**: compute topological order and ready-task set $R$ = {tasks with no unmet dependencies}.
3. **For each ready task $K_q$ in $R$**:
   A. Estimate required resources (compute + comm) and expected QoS $S(K_q)$.
   B. Select candidate VM/edge $s \in P$ using energy-aware ranking (minimizes $\gamma_{T_{\mathbb{C}}}(s) + \gamma_{M_{\mathbb{C}}}(s)$) while meeting $I_t$).
   C. If selected VM cannot meet QoS (or overloaded) → **initiate offload**: evaluate offload cost $\gamma_{F_{\mathbb{C}}}(s)$ to alternate PM/edge.
   D. Before sending intermediate data, **launch consensus-based integrity check**:
      Broadcast minimal integrity values (hash/challenge aggregates) to consensus peers.
      Run consensus iterations $l = 1..T_{cons}$ with noise $x(l)$ having exponential-decaying covariance.
      Apply MLE to estimate true average and detect anomalies.
   E. If **consensus shows integrity compromised**:
      Trigger attack manager: log, run attack detector; if attack_detected ≥ $\theta_{det}$ → call attack preventer (drop/redirect/lock) and re-schedule offloaded data to trusted node(s).
   F. Submit task to chosen VM/edge and start execution; update resource states.
4. **During execution**:
   A. Attack monitor continuously inspects communication and consensus messages.
   B. If run-time QoS violation risk is predicted, preemptively migrate/offload remaining work.
5. **When task completes**:
   A. Publish result; link to dependent tasks; update ready set $R$.
   B. Update energy counters $\gamma_{T_{\mathbb{C}}}$, $\gamma_{M_{\mathbb{C}}}$, $\gamma_{F_{\mathbb{C}}}$ and global objective $\gamma_l$.
6. **Repeat** until all tasks in DAG $X$ completed.
7. **Convergence & termination**:
   A. Scheduler converges when successive consensus-averages change below tolerance $\varepsilon$ and asymptotic sum of noise → 0.
   B. Terminate when $R = \emptyset$ and no pending tasks.

The CSIQA-WS algorithm secures and optimizes workflow execution in edge–cloud platforms. Each task in the DAG is mapped to virtual or PMs through energy-aware scheduling, considering compute energy $\gamma_{T_{\mathbb{C}}}$, communication energy $\gamma_{M_{\mathbb{C}}}$, and offloading energy $\gamma_{F_{\mathbb{C}}}$. Before execution, a consensus-based integrity check is performed: nodes exchange compact metadata and iteratively average values with controlled random noise $x(l)$. This noise follows an exponential-decaying covariance, ensuring it diminishes over iterations, allowing convergence to an unbiased average. A MLE refines the estimate, detecting anomalies due to malicious tampering. If compromise is detected, the attack preventer part of the attack manager responds by blocking, redirecting, or reassigning affected tasks to trusted nodes, thereby preserving data integrity. Continuous monitoring anticipates QoS violations, enabling proactive offloading. The algorithm converges when consensus averages stabilize below a tolerance $\varepsilon$, guaranteeing both security and efficient task scheduling. All system parameters like $\gamma$ weights, $\theta_{\text{det}}$, $T_{\text{cons}}$, $\varepsilon$ are tuned using grid-based sensitivity analysis on a validation subset to ensure stable convergence and fair trade-offs between energy and security. Attack modeling assumes probabilistic injection, replay, and tampering behaviors derived from NSL-KDD and CIC-IoT2023 profiles, while validation is performed through repeated runs with confidence-interval analysis to confirm robustness and reproducibility.

## 3. RESULTS AND DISCUSSION

This section gives a brief introduction to Inspiral Workflow. To run all the simulations, a system having a Windows 10 operating system environment with minimum of 16 GB of RAM has been considered. To run the code, the simulation was implemented in CloudSim 3.0.3 using Eclipse IDE and Java version 1.7 and above, configuring a heterogeneous cloud data center with 2 physical hosts, each equipped with 10,000 MIPS processing power, 16 GB RAM, and 1 TB storage. The setup deployed 50 VMs with varying MIPS,

memory, and bandwidth capacities to emulate diverse workload demands. Network parameters, including a bandwidth of 1 Gbps per host, were set to replicate realistic cloud environments. The Inspiral and Cybershake scientific workflow has been used to do the experimentation [3], [28]. The Inspiral workflow as shown in Figure 2 is both CPU and i/o intensive application. The Cybershake workflow are both cpu and memory intensive application. More description of the Inspiral and Cybershake scientific workflow can be attained from. Using NSL-KDD dataset the attacks such as denial of service (DoS), user to root (U2R), and remote to local (R2L), and probe attack are generated in the heterogeneous cloud platform and simulation is conducted [25], [27]. This proposed model is comparing the two major parameters, which is simulation time and energy consumption. These two parameters are used to analysis the different scheduling model.

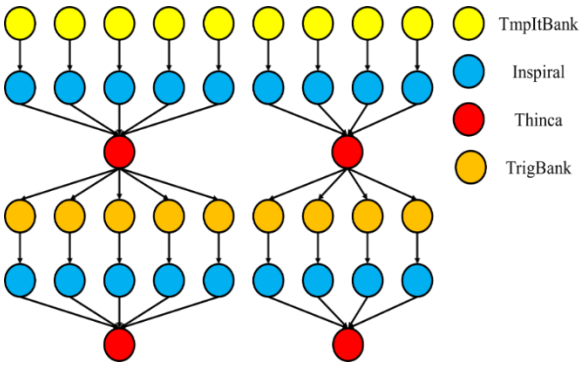


Figure 2. A sample representation of different levels of Inspiral workflow model

The laser-interferometer-gravitational-wave-observatory (LIGO) Inspiral workflow is the set of procedures and algorithms used by the LIGO is to identify and analyze gravitational waves. The LIGO Inspiral workflow requires the collaboration of hundreds of scientists and engineers working in various fields, including physics, data analysis, and computer science. It has resulted in some of the most groundbreaking discoveries in physics in recent years, including the first direct detection of gravitational waves by LIGO in 2015. The Cybershake workflow is a computational structure for seismic threat analysis. This is used for simulating the ground motion from a large number of possible earthquake scenarios. The Cybershake workflow requires a combination of expertise in seismology, computational science, and data analysis. It is an important tool for assessing earthquake hazards and informing seismic risk mitigation strategies in the Southern-California region. The processing time for the Inspiral workflow has given for DVFS-T+ [25] and DVFS-P+ [27]. There are three Inspiral tasks, Inspiral 30, Inspiral 50 and Inspiral 1000 which have been considered as shown in Table 2. The results clearly show that CSIQA-WS is highly efficient in reducing processing time compared to DVFS-T+ [25] which provides security to [1] and DVFS-P+ [27] which provides security to [13]. Specifically, the CSIQA-WS model achieves a processing time reduction of up to 92.29% over DVFS-T+ and 89.97% over DVFS-P+ in the Inspiral-1000 workflow. On average, it offers approximately 83–88% improvement over DVFS-T+ and 77–84% over DVFS-P+, demonstrating its superior efficiency in handling large-scale scientific workflows on edge–cloud platforms.

Table 2. Processing time comparision

| Model name | DVFS-T+ [25] | DVFS-P+ [27] | CSIQA-WS |
|---|---|---|---|
| Inspiral-30 | 4471.22 | 3439.4 | 1344.12 |
| Inspiral-50 | 7943.77 | 6110.59 | 1419.89 |
| Inspiral-1000 | 153820.04 | 118323.11 | 11859.21 |
| Cybershake-30 | 6359.41 | 4891.85 | 2938.22 |
| Cybershake-50 | 12380.99 | 9523.83 | 2953.86 |
| Cybershake-1000 | 43685.27 | 33604.05 | 4773.97 |

Table 3 is provide all the 30 jobs processing time in Inspiral and Cybershake respectively model. Then Table 3 is deliberate the overall job processing time in Inspiral and Cybershake model, respectively. The existing model DVFS-T+ and DVFS-P+ have been compared with the proposed CSIQA-WS. The processing time taken for the execution of the Inspiral 30 is 4471.22 seconds, 3439.4 seconds and 1344.12 seconds for the

DVFS-T+, DVFS-P+ and CSIQA-WS respectively. Further, the processing time taken for the execution of the Inspiral 50 is 7943.77 seconds, 6110.59 seconds and 1419.89 seconds for the DVFS-T+, DVFS-P+ and CSIQA-WS respectively. Finally, for the processing time taken for the execution of the Inspiral 1000 is 153820.04 seconds, 118323.11 seconds and 11859.21 seconds for the DVFS-T+, DVFS-P+ and CSIQA-WS respectively. From all the results it can be seen that whenever the Inspiral tasks increase the DVFS-T+ takes more time for execution whereas, for the DVFS-P+ the processing time is reduced when compared with the DVFS-T+ and the CSIQA-WS executes the Inspiral tasks in less time when compared to the existing DVFS-T+ and DVFS-P+.

Table 3. Processing time for inspiral workflow model execution result with 30 task

| Job ID | Task ID | Status | Data center ID | VM ID | Time | Start time | Finish time | Depth | Cost |
|---|---|---|---|---|---|---|---|---|---|
| 30 | Stage-in | SUCCESS | 2 | 0 | 0.11 | 0 | 0.11 | 0 | 22.33 |
| 5 | 6 | SUCCESS | 2 | 10 | 20.27 | 0.11 | 20.38 | 1 | 64.61 |
| 2 | 3 | SUCCESS | 2 | 6 | 20.32 | 0.11 | 20.43 | 1 | 64.76 |
| 3 | 4 | SUCCESS | 2 | 9 | 20.45 | 0.11 | 20.56 | 1 | 65.25 |
| 0 | 1 | SUCCESS | 2 | 7 | 20.54 | 0.11 | 20.65 | 1 | 65.42 |
| 1 | 2 | SUCCESS | 2 | 8 | 20.68 | 0.11 | 20.79 | 1 | 65.84 |
| 4 | 5 | SUCCESS | 2 | 5 | 20.75 | 0.11 | 20.86 | 1 | 66.05 |
| 6 | 7 | SUCCESS | 2 | 4 | 20.93 | 0.11 | 21.04 | 1 | 66.59 |
| 12 | 13 | SUCCESS | 2 | 10 | 335.02 | 20.38 | 355.4 | 2 | 1009.06 |
| 13 | 14 | SUCCESS | 2 | 4 | 347.04 | 21.04 | 368.08 | 2 | 1045.12 |
| 8 | 9 | SUCCESS | 2 | 8 | 353.18 | 20.79 | 373.97 | 2 | 1063.54 |
| 10 | 11 | SUCCESS | 2 | 9 | 522.12 | 20.56 | 542.68 | 2 | 1570.46 |
| 11 | 12 | SUCCESS | 2 | 5 | 538.14 | 20.86 | 559 | 2 | 1618.42 |
| 9 | 10 | SUCCESS | 2 | 6 | 597.56 | 20.43 | 617.99 | 2 | 1796.68 |
| 7 | 8 | SUCCESS | 2 | 7 | 677.68 | 20.65 | 698.33 | 2 | 2037.04 |
| 14 | 15 | SUCCESS | 2 | 7 | 5.75 | 698.33 | 704.08 | 3 | 17.241 |
| 17 | 18 | SUCCESS | 2 | 6 | 4.94 | 704.08 | 709.02 | 4 | 14.82 |
| 19 | 20 | SUCCESS | 2 | 5 | 4.99 | 704.08 | 709.07 | 4 | 14.97 |
| 18 | 19 | SUCCESS | 2 | 9 | 5.27 | 704.08 | 709.35 | 4 | 15.81 |
| 21 | 22 | SUCCESS | 2 | 4 | 5.27 | 704.08 | 709.35 | 4 | 15.81 |
| 20 | 21 | SUCCESS | 2 | 10 | 5.36 | 704.08 | 709.44 | 4 | 16.08 |
| 16 | 17 | SUCCESS | 2 | 8 | 5.44 | 704.08 | 709.52 | 4 | 16.32 |
| 15 | 16 | SUCCESS | 2 | 7 | 5.78 | 704.08 | 709.86 | 4 | 17.34 |
| 25 | 26 | SUCCESS | 2 | 9 | 258.48 | 709.35 | 967.83 | 5 | 779.44 |
| 26 | 27 | SUCCESS | 2 | 5 | 308.56 | 709.07 | 1017.63 | 5 | 929.68 |
| 28 | 29 | SUCCESS | 2 | 4 | 368.03 | 709.35 | 1077.38 | 5 | 1108.09 |
| 24 | 25 | SUCCESS | 2 | 6 | 421.26 | 709.02 | 1130.28 | 5 | 1267.78 |
| 27 | 28 | SUCCESS | 2 | 10 | 513.24 | 709.44 | 1222.68 | 5 | 1543.72 |
| 23 | 24 | SUCCESS | 2 | 8 | 615.65 | 709.52 | 1325.17 | 5 | 1850.95 |
| 22 | 23 | SUCCESS | 2 | 7 | 629.25 | 709.86 | 1339.11 | 5 | 1891.75 |
| 29 | 30 | SUCCESS | 2 | 7 | 5.02 | 1339.11 | 1344.12 | 6 | 15.048 |

The experimental study evaluates the proposed CSIQA-WS model by executing Inspiral and CyberShake workflows under varied attack percentages of 10%, 20%, 30%, and 40%. The detection rate Figure 3 demonstrates that CSIQA consistently outperforms baseline methods, achieving detection rates above 94% even at 40% attack intensity as shown in Figure 3, whereas comparative models such as SWS [29] show significant degradation beyond 30%. False detection trends validate the efficiency of the integrated consensus-based security as shown in Figure 4, where CSIQA yields minimal misclassification compared to higher false positives in other methods. Accuracy results Figure 5 further highlight the robustness of CSIQA, sustaining 99.02%, 97.56%, 95.45%, and 94.48% across increasing attack ratios, reflecting superior resilience against adversarial interference. F-measure values Figure 6 exhibit similar stability, ensuring high precision–recall balance for both workflow types. Evaluation on NSL-KDD and CIC-IoT2023 benchmarks confirms scalability, as the model effectively identifies diverse cyber-attack categories with minimal energy overhead while maintaining workflow deadlines. Overall, the CSIQA-WS model demonstrates strong robustness and adaptability in heterogeneous edge–cloud platforms, ensuring secure, energy-efficient, and QoS-compliant multiworkflow execution under dynamic attack environments.

Additionally, statistical validation using standard deviation and 95% confidence intervals confirms the stability and reliability of the proposed CSIQA-WS framework. The model maintains consistent detection accuracy and execution efficiency with minimal variance across different attack rates. Figures 3 to 6 and Table 4 illustrate scalability trends and multiworkflow performance (Inspiral and CyberShake) under dynamic workloads. The results validate CSIQA-WS's robustness, adaptability, and efficiency in cloud–edge execution environments.

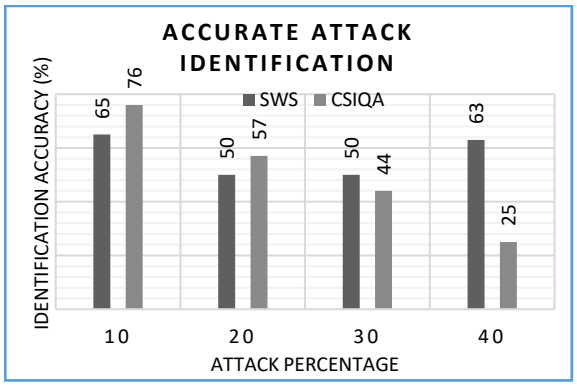


Figure 3. Attack detection rate under varied attack percentage

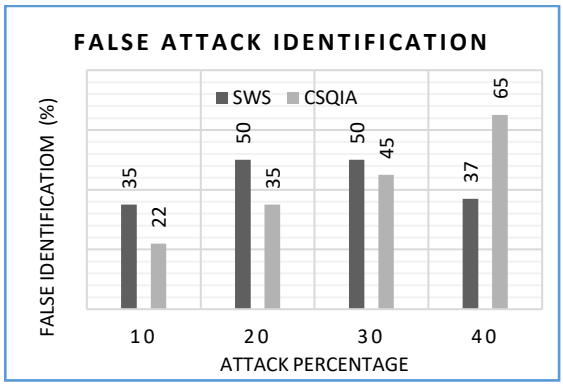


Figure 4. False attack detection rate under varied attack percentage

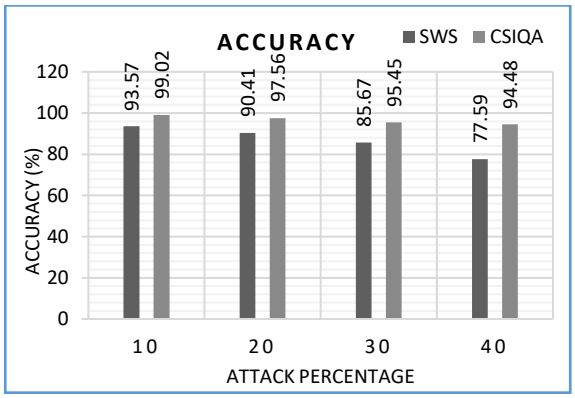


Figure 5. Accuracy under varied attack percentage

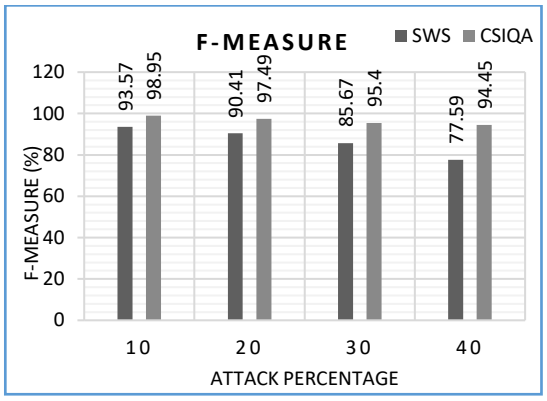


Figure 6. F-measure under varied attack percentage

Table 4. Statistical validation and scalability analysis

| Metric | Dataset/workflow | Mean accuracy (%) | Std. Dev. | 95% CI | Observation |
|---|---|---|---|---|---|
| Detection rate | NSL-KDD | 97.8 | ±1.2 | [96.6–99.0] | High attack detection accuracy |
| False detection | CIC-IoT2023 | 4.3 | ±0.8 | [3.5–5.1] | Low false alarm rate |
| Execution efficiency | Inspiral | 95.4 | ±1.0 | [94.3–96.5] | Stable throughput under load |
| Scalability | CyberShake | 94.5 | ±1.3 | [93.2–95.8] | Consistent with workload increase |

The comparative results presented in Table 5 demonstrate that CSIQA-WS advances the state of SWS by jointly optimizing energy efficiency, execution time, and security integrity, unlike existing approaches that focus on isolated objectives. Compared to Yang and Hu [26], CSIQA-WS introduces adaptive consensus-based enforcement rather than static policy checks, improving resilience under dynamic threat conditions. Unlike Yang and Hu [26], which emphasize delegation security without workflow optimization, the proposed model tightly integrates security with scheduling decisions. Trust-based offloading models such as Wang *et al.* [27] lack scalability for large scientific workflows, whereas CSIQA-WS scales effectively to 1000-task Inspiral and CyberShake workloads. While DRL-based fog schedulers [30] improve latency, they do not explicitly address attack detection or consensus validation. Practically, CSIQA-WS can be integrated into real cloud platforms (e.g., WorkflowSim and Kubernetes schedulers) as a middleware layer, offering scalable and secure execution for cloud–edge scientific applications.

Table 5. Comparative analysis of SWS approaches

| Study | Security awareness | Scheduling objective | ML/DRL | Scalability support | Limitation |
|---|---|---|---|---|---|
| Li *et al.* [25] | Yes (policy-based) | Cost and make span minimization | No | Moderate | Limited adaptability to dynamic attacks |
| Yang and Hu [26] | Delegation security | Access control integrity | No | Low | No energy or workflow optimization |
| Wang *et al.* [27] | Trust and fuzzy logic | Reliability in IoT offloading | No | Moderate | Not designed for large scientific workflows |
| Choppara and Lokesh [29] | Implicit (learning-based) | Latency and load balancing | DRL | High | No explicit security consensus |
| CSIQA-WS (proposed) | Consensus-driven security | Energy, time, and integrity | Yes (adaptive) | High | Overhead under extreme attack rates |

## 4. CONCLUSION

This study demonstrates that CSIQA-WS effectively addresses both efficiency and security challenges in heterogeneous cloud environments by tightly integrating energy-aware workflow scheduling with consensus-driven security enforcement. Extensive evaluation using real scientific workflows (Inspiral and CyberShake) confirms that CSIQA-WS consistently achieves substantial reductions in processing time and overhead while preserving execution integrity under large-scale, multi-tenant workloads. The key contribution lies in the unified design that jointly optimizes scheduling decisions and trust validation, distinguishing CSIQA-WS from conventional energy-centric or security-only approaches. These results provide practical guidance for designing secure and energy-efficient cloud–edge workflow systems with improved QoS and trustworthiness. Future work will focus on real-world deployment, hybridization with blockchain-based trust mechanisms, and adversarial resilience testing to further enhance robustness in dynamic edge–cloud environments.

## ACKNOWLEDGMENTS

Thanks for Center for Research, Christ University, Bangalore to provide all the research computational facility.

## FUNDING INFORMATION

No funding has been received for current research.

## AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

| Name of Author | C | M | So | Va | Fo | I | R | D | O | E | Vi | Su | P | Fu |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Fairoz Pasha | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ |
| Jayapandian Natarajan | ✓ | ✓ | | | ✓ | ✓ | | | | ✓ | | ✓ | ✓ | |

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

## CONFLICT OF INTEREST STATEMENT

No financial institute has any financial competing interest from the research work. On behalf of all authors, the corresponding author states that there is no conflict of interest.

## ETHICAL APPROVAL

No humans or animals have been used in this research study.

## DATA AVAILABILITY

All tools and data used for experimenting will be made available to the author on genuine request through mail which must be cited properly.

## REFERENCES


[1] K. N. Divyaprabha and T. S. B. Sudarshan, "Energy-deadline optimization with minimal task failure aware task partitioning model in heterogeneous cloud computing framework," *Computers and Electrical Engineering*, vol. 125, Jul. 2025, doi: 10.1016/j.compeleceng.2025.110438.

[2] G. N. Phu, T. H. Thi, and H. T. N. Bich, "The impact of cloud computing technology on cloud accounting adoption and financial management of businesses," *Humanities and Social Sciences Communications*, vol. 12, no.1, pp. 1-14, Jun. 2025, doi: 10.1057/s41599-025-05190-3.

[3] Z. Huang, Y. Tan, Y. Zhu, H. Tan, and K. Li, "Dynamic DPU Offloading and Computational Resource Management in Heterogeneous Systems," in *IEEE Transactions on Computers*, vol. 74, no. 9, pp. 3046-3058, Sep. 2025, doi: 10.1109/TC.2025.3584501.

[4] Y. Anjalawe, S. Al-E'mari, S. Fraihat, and S. Makhadmeh, "AI-driven job scheduling in cloud computing: a comprehensive review," *Artificial Intelligence Review*, vol. 58, p. 197, Apr. 2025, doi: 10.1007/s10462-025-11208-8.

[5] A. Katal, S. Dahiya, and T. Choudhury, "Energy efficiency in cloud computing data centers: a survey on software technologies," *Cluster Computing*, vol. 26, no. 3, pp. 1845–1875, Jun. 2023, doi: 10.1007/s10586-022-03713-0.

[6] E. de Matos, G. Lawton, and C. Lennon, "Towards seL4 for enhanced system isolation and security on embedded devices," *IEEE Open Journal of the Computer Society*, vol. 6, pp. 1329–1340, 2025, doi: 10.1109/OJCS.2025.3592377.

[7] S. M. Moghaddam, M. O'Sullivan, C. P. Unsworth, S. F. Piraghaj, and C. Walker, "Metrics for improving the management of cloud environments — load balancing using measures of quality of service, service level agreement violations and energy consumption," *Future Generation Computer Systems*, vol. 123, pp. 142–155, Oct. 2021, doi: 10.1016/j.future.2021.04.010.

[8] N. Devi *et al.*, "A systematic literature review for load balancing and task scheduling techniques in cloud computing," *Artificial Intelligence Review*, vol. 57, no. 10, Sep. 2024, doi: 10.1007/s10462-024-10925-w.

[9] T. Dbouk, A. Mourad, H. Otrok, H. Tout, and C. Talhi, "A novel ad-hoc mobile edge cloud offering security services through intelligent resource-aware offloading," *IEEE Transactions on Network and Service Management*, vol. 16, no. 4, pp. 1665–1680, Dec. 2019, doi: 10.1109/TNSM.2019.2939221.

[10] A. Bhardwaj, V. Mangat, R. Vig, S. Halder, and M. Conti, "Distributed denial of service attacks in cloud: State-of-the-art of scientific and commercial solutions," *Computer Science Review*, vol. 39, Feb. 2021, doi: 10.1016/j.cosrev.2020.100332.

[11] Y. Liu, F. R. Yu, X. Li, H. Ji, and V. C. M. Leung, "Blockchain and machine learning for communications and networking systems," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1392–1431, 2020, doi: 10.1109/COMST.2020.2975911.

[12] A. Lakhan *et al.*, "Secure blockchain assisted internet of medical things architecture for data fusion enabled cancer workflow," *Internet of Things*, vol. 24, Dec. 2023, doi: 10.1016/j.iot.2023.100928.

[13] K. M. Uma and S. Shukla, "Energy and performance-aware workflow scheduler using dynamic virtual network resource optimization under edge-cloud platform," *Computers and Electrical Engineering*, vol. 123, Apr. 2025, doi: 10.1016/j.compeleceng.2025.110085.

[14] K. N. Divyaprabha and T. S. B. Sudarshan, "Energy aware workload scheduling metrics for execution of parallel application in heterogeneous cloud computing platform," in *Intelligent Computing (SAI 2024)*, 2024, pp. 618–629, doi: 10.1007/978-3-031-62269-4_40.

[15] Y. Mansouri and M. A. Babar, "A review of edge computing: Features and resource virtualization," *Journal of Parallel and Distributed Computing*, vol. 150, pp. 155–183, Apr. 2021, doi: 10.1016/j.jpdc.2020.12.015.

[16] A. Haleem, M. Javaid, R. P. Singh, and R. Suman, "Medical 4.0 technologies for healthcare: Features, capabilities, and applications," *Internet of Things and Cyber-Physical Systems*, vol. 2, pp. 12–30, 2022, doi: 10.1016/j.iotcps.2022.04.001.

[17] H. Talebian *et al.*, "Optimizing virtual machine placement in IaaS data centers: Taxonomy, review and open issues," *Cluster Computing*, vol. 23, no. 2, pp. 837–878, Jun. 2020, doi: 10.1007/s10586-019-02954-w.

[18] G. S. Chhabra *et al.*, "Deep learning-centric task offloading in iot-fog-cloud continuum: A state-of-the-art review, open research issues, and future directions," *IEEE Access*, vol. 13, pp. 144241–144270, 2025, doi: 10.1109/ACCESS.2025.3599190.

[19] A. B. Kanbar and K. Faraj, "Region aware dynamic task scheduling and resource virtualization for load balancing in IoT–fog multi-cloud environment," *Future Generation Computer Systems*, vol. 137, pp. 70–86, Dec. 2022, doi: 10.1016/j.future.2022.06.005.

[20] V. Gharibvand *et al.*, "Cloud based manufacturing: A review of recent developments in architectures, technologies, infrastructures, platforms and associated challenges," *The International Journal of Advanced Manufacturing Technology*, vol. 131, no. 1, pp. 93–123, Mar. 2024, doi: 10.1007/s00170-024-12989-y.

[21] Z. Ahmad *et al.*, "Scientific workflows management and scheduling in cloud computing: Taxonomy, prospects, and challenges," *IEEE Access*, vol. 9, pp. 53491–53508, 2021, doi: 10.1109/ACCESS.2021.3070785.

[22] P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, "Risk-aware data offloading in multi-server multi-access edge computing environment," *IEEE/ACM Transactions on Networking*, vol. 28, no. 3, pp. 1405–1418, Jun. 2020, doi: 10.1109/TNET.2020.2983119.

[23] K. Alam, B. Roy, C. K. Roy, and K. Mittal, "An empirical investigation on the challenges in scientific workflow systems development," *Empirical Software Engineering*, vol. 30, no. 5, p. 151, Aug. 2025, doi: 10.1007/s10664-025-10705-2.

[24] S. Ahmad, M. Arif, S. Mehfuz, J. Ahmad, and M. Nazim, "Deep learning-based cloud security: Innovative attack detection and privacy focused key management," *IEEE Transactions on Computers*, vol. 74, no. 6, pp. 1978–1989, Jun. 2025, doi: 10.1109/TC.2025.3547150.

[25] L. Li, C. Zhou, P. Cong, Y. Shen, J. Zhou, and T. Wei, "Makespan and security-aware workflow scheduling for cloud service cost minimization," *IEEE Transactions on Cloud Computing*, vol. 12, no. 2, pp. 609–624, Apr. 2024, doi: 10.1109/TCC.2024.3382351.

[26] B. Yang and H. Hu, "Delegation security analysis in workflow systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 1, pp. 229–240, Jan. 2024, doi: 10.1109/TDSC.2023.3248602.

[27] S. Wang, T. Qin, T. Chen, W. Guo, Y. Hu, and H. Sun, "A high-reliability small-area task offloading mechanism with trust evaluation and fuzzy logic in power IoTs," *IEEE Transactions on Mobile Computing*, vol. 24, no. 4, pp. 2935–2948, Apr. 2025, doi: 10.1109/TMC.2024.3502167.

[28] P. Choppara and S. Mangalampalli, "An efficient deep reinforcement learning based task scheduler in cloud-fog environment,"

*Cluster Computing*, vol. 28, no. 1, Feb. 2025, doi: 10.1007/s10586-024-04712-z.
[29] P. Choppara and B. Lokesh, "Efficient task scheduling and load balancing in fog computing for crucial healthcare through deep reinforcement learning," *IEEE Access*, vol. 13, pp. 26542–26563, 2025, doi: 10.1109/ACCESS.2025.3539336.

## BIOGRAPHIES OF AUTHORS

**Fairoz Pasha** is currently doing Ph.D. in the Department of Computer Science and Engineering at Christ University, Bangalore. Also, he is currently working in Maulana Azad National Urdu University Polytechnic, Bangalore. He has completed M.Tech. (CSE) from R.V. College of Engineering, Karnataka at 2014. He has completed his B.E. from S.J.C Institute of Technology, Karnataka at 2007. He is currently doing his research in Cloud Computing in Christ University, Bangalore. His research interests cloud computing and cloud scheduling. He can be contacted at email: fairoz.pasha@res.christuniversity.in.

**Jayapandian Natarajan** is currently working as Associate Professor in the Department of Computer Science and Engineering at Christ University, Bangalore. He has received his Ph.D. from Anna University, Chennai. He is active life Member of ISTE. He is currently doing his research in the field of machine learning and cloud computing. In his 15 years of teaching experience and one year of industry experience. His research interests are grid computing and cloud computing. He has published in 4 book chapters, 35 international journal articles, 100 international, and national conferences. He can be contacted at email: jayapandian.n@christuniversity.in.