Journal of Computer Applications and Information Technology
Vol. 1, Issue. 3, Sep 2025, pp. 47~59
ISSN: 3107-6424

Scalable and Secure Framework for Cloud-Based IT Infrastructure Using Software-Defined Networking and Distributed Storage

Deepika Bandhana[1], Sarvesh Chand[2]
[1]School of Science and Technology, The University of Fiji, Lautoka, Fiji
deepikabandhana@yahoo.com
[2]Pacific Technical and Further Education (Pacific TAFE),
The University of The South Pacific, Suva, Fiji.
sarvesh.chand@usp.ac.fj

Article Info

Article History:

Received Apr 13, 2025
Revised May 10, 2025
Accepted Jun 12, 2025

Keywords:

Cloud Computing
Software-Defined Networking
IT Infrastructure
Cloud Service Provider
Virtual Machine

ABSTRACT

Due to its scalable and affordable processing, cloud computing (CC) has become the real trend in enterprise IT service models. The challenge of maintaining and safeguarding hybrid cloud infrastructures has increased as more businesses use them. Software-defined networking (SDN) is being adopted by enterprise networks in order to reduce operational costs and provide flexibility in network management. Both technologies are frequently used in enterprise information technology (IT) services. However, it's uncertain how software-defined networking and CC will affect the security of commercial networks. Advanced computing paradigms like serverless computing, containerization, and edge computing are supported by the smooth administration of compute, storage, and networking resources made possible by SDI's interaction with cloud platforms. The study presents a Scalable and Secure Framework for Cloud-Based IT Infrastructure Using Software-Defined Networking and Distributed Storage (SFCIT-SDNDS). The SFCIT-SDNDS technique involves data collection, and module development processes. SDN will identify any threats in the various domains that make up the network and send the encrypted data to the blockchain. Our SFCIT-SDNDS system can use real-world network data to indicate a wide range of applications, as can our new network paradigm. A set of experiment was performed to analyze the scalability of the SFCIT-SDNDS model. According to our comprehensive experimental examination of performance evaluation, the suggested SFCIT-SDNDS framework outperforms other existing approaches by 97.75%.

Corresponding Author:

Deepika Bandhana,
School of Science and Technology,
The University of Fiji, Lautoka, Fiji.
Email: deepikabandhana@yahoo.com

1. INTRODUCTION

Cloud computing (CC) is a new technology that gives users access to physical resources via virtualization technology [1]. The advantage of enabling network access to a scalable combination of shared virtual or physical resources with on-demand available and self-owned service provisioning services is driving the growth of the CC sector. Because of these benefits, the number of people embracing CC has also increased. However, as CC has grown, security threats have started to emerge [2]. As more and more businesses use CC to improve scalability, streamline operations, and cut costs, cloud networking security has emerged as a crucial component of contemporary IT infrastructures. Businesses can store data, run apps, and use computer resources whenever they need them thanks to the flexibility and efficiency that cloud environments offer [3]. But as more businesses move to cloud-based architecture, they must deal with the difficult challenge of maintaining strong security across an increasing number of different platforms. The development of hybrid on premise and cloud infrastructures, which incorporate both private and public cloud systems with traditional on premise networks, makes this difficulty even more apparent.

Virtualization technologies are being used to study a number of cloud security measures, which makes it challenging to apply existing digital forensic techniques [4]. In Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS) settings, access to specific system levels is limited based on the service model governing that layer, which is controlled by the Cloud Service Provider (CSP). As a result, it makes sense to provide the CSP with the log information produced at the inaccessible layer via agreement. In traditional digital forensics, investigators have exclusive control over the evidence [5]. However, in a cloud context, data centers are spread out geographically; cloud service customers (CSCs) share virtual networks, auto scaling, load balancing, unreliable data, and physical infrastructure that vanishes when the instance is shut down in order to create a seamless service environment. Since it is impossible for the investigator to directly capture the data and collect the data from the remote server, it is crucial to ensure the veracity of the log data in addition to recording it for cloud forensics prior to a security incident for inquiry [6].

SDN is a revolutionary method to network design and deployment that emphasizes separating the control of network function (NF) from networking hardware (switches, routers, firewalls, load balancers, etc.). The open flow protocol enables software-defined networking (SDN) switches to meet the traditionally fulfilled functionalities through a physical device multitude by leveraging the flexibility offered by the ability to access header information from various open system interconnections (OSI) stack layers [7]. Because SDN programmable network interfaces were included, SDN was a great platform for multi-tenant data center deployments that required dynamism and flexibility. This was especially true in the infrastructure-as-a-service (IaaS) paradigm, where renters kept virtual machines (VMs) in order to have financial and technological flexibility. Cloud service providers (CSP), cyber security, and data center administrators are all primarily concerned with privacy and security [8].

By offering firewalls, network segmentation, virtual private networks (VPNs), IDS/IPS, network monitoring, traffic shaping, and load balancers to guard against different security risks, the cloud network infrastructure is essential to guaranteeing the security of cloud resources in data center. A networking design known as SDN divides a network's control plane from its data plane. In conventional networking, network devices like switches and routers carry out both tasks since the control plane and data plane are closely connected. While the data plane is in charge of actually

sending the traffic, the control plane is in charge of deciding how it should be routed across the network [9]. Each user or tenant would have their own physical network infrastructure in traditional networking, which might be expensive and challenging to maintain. In conventional networking, each tenant or user would have their own physical network infrastructure, which might be costly and difficult to manage. SDN enables the centralized controller to keep an eye on everything, increasing application flexibility and efficiency.

In order to defend against cyberattacks, security policies and controls may be quickly deployed and reconfigured because to SDN's dynamic nature. Because SDN is dynamic, security policies and controls may be swiftly implemented and rearranged to protect against cyberattacks. By flooding the virtual machine (VM) hosting the services with multiple packet volumes, the most basic volumetric assault targets Cloud resources and disrupts the quality of experience (QoE) for regular users, either completely or partially.

This research paper presents a cloud SDN security architecture and implements a security model with an attack-detecting technique in the data plane and mitigating control in the SDN control plane. Experiments demonstrate that the cooperative security strategy in SDN hardly slightly alters processing costs. Additionally, this strategy guards against middle-box appliances and downstream servers, prevents the control-plane from becoming saturated, and prevents flow-table/miss assaults in the network. In order to identify coarse-grained attacks and take corrective action, switches must be permitted to process new packets in the data plane. Data plane security monitoring and control plane threat analysis are features of an SDN-integrated cloud management system [10]. These method assessments demonstrate that, in comparison to traditional perimeter/firewall solutions, the expandable data plane of stately SDN in the model with the NF service chain provides higher security. Presenting a new and adaptable architecture for creating a safe SDN–IoT-based blockchain network is one of this article's main contributions. The following is a summary of the contributions.

- ❖ We create and build a new and adaptable architecture for IT domains that is made up of SDN controllers and CC services connected by a safe IT infrastructure.
- ❖ In order to detect assaults in the SDN network, we utilize Mininet1 to create the SDN network. Additionally, a method for identifying attacks is provided, wherein IoT domains are assigned a threshold value. The CC network is in line with the SDN network to thwart attacks.
- ❖ We assess potential threats that could cause a single point of failure on the SDN controller.

In this article, we present a Scalable and Secure Framework for Cloud-Based IT Infrastructure Using Software-Defined Networking and Distributed Storage (SFCIT-SDNDS). The SFCIT-SDNDS technique involves data collection, and module development processes. SDN will identify any threats in the various domains that make up the network and send the encrypted data to the blockchain. Our SFCIT-SDNDS system can use real-world network data to indicate a wide range of applications, as can our new network paradigm. A set of experiment was performed to analyze the scalability of the SFCIT-SDNDS model.

2. RELATED WORKS

Razvan and Mitica [11] suggests peer-to-peer blockchain storage, evidence collection, and forensic structure design. Unauthorized users will be protected by the Secure Block Verification

Mechanism (SBVM). The cuckoo search optimization approach is used to generate secret keys. The cloud authentication server stores and encrypts all data for privacy. Algebraically Based on Confidentiality Cryptosystem learning is given a new encryption technique called homomorphism. The SDN controller assigns a block to each piece of data, and the history is stored as metadata about the data. Every block has a 512-bit hash-based tree with Secure Hash Algorithm version. Our method tracks user data (GNNSC) in Smart Contracts using graph neural networks based on graph theory. Lastly, evidence analysis is made possible via a blockchain-based evidence graph. The studies were conducted using network simulator 3.30 (for software-defined networks) and Python in a cloud environment. With our recently discussed forensic construction using blockchain (FAuB), we were able to achieve good results.

Jalali and Reed [12] emphasizes the role of SDN in this context, especially through the use of Transport Layer Security (TLS) encryption, and emphasizes the vital significance of security in cloud and big data contexts. SDN controllers and switches may communicate securely thanks to TLS. Although earlier research has identified possible delay effects of TLS, this work offers conflicting results in the context of SDN switches that are frequently utilized in cloud architecture. This paper demonstrates through a thorough analysis the critical impact TLS plays in bolstering the security and integrity of SDN connections. SDN networks, which are essential to cloud infrastructures and big data applications, can be made more secure with the help of technical insights into TLS integration, its effect on performance, and real-world case studies.

Vimal et al. [13] discusses the stability routing protocol, which makes the assumption that devices and packet flow are credible. Establish trust between network element flows and energy or Quality of Service (QoS) conditions in order to construct dependable Software-Defined Network (SDN) routes. The Cognitive Protocol Network (CPN) uses the SDN architecture to boost energy levels. Decentralized choices are made via Stochastic Neural Networks (SNNs), which are trained using data taken from perceptual packets. In order to improve IoT encryption for information access control, the suggested network infrastructure is created and included into SerIoT approaches. The technology's adaptability allows it to get over the system's erratic connectivity and node declines in terms of deterministic energy, target node, restricted interval, and prospective cryptographic capacity. Proper marketing creates an end-to-end antitheft solution that satisfies a set of preset circuit limits based on genuine statistical data. Because clusters are clearly unstable, a study must cooperate by showcasing a number of defects, which is crucial for the platform's effectiveness.

Van Anh et al. [14] present a new framework that improves edge computing capabilities by utilizing software-defined networking (SDN). This framework, which was created specially to support real-time Internet of Health Things applications, should enable dynamic and flexible communication between edge and cloud servers. It is anticipated that the system will enable load balancing, network optimization, and resource usage in an effective manner by connecting servers to the Software-Defined Networking controller. Consequently, this makes it possible to deliver healthcare treatments in real time. Finally, the influence of the proposed framework on service response time is examined in order to evaluate its effectiveness. The study's findings show that the suggested framework greatly improves the service response times of IoHT systems for a range of traffic and workloads.

Thangavel et al. [15] investigates particular SDN-based methods for adaptive QoS, dynamic load, and flow control. The research's findings are given with an emphasis on programmability, automated control, and scalability, highlighting the advantages of employing SDN opportunistic

architectures over traditional Network Systems. Furthermore, we demonstrate how AI and ML may be applied at the SDN controller level to enhance predictive analytics and/or real-time traffic decision-making. Therefore, we use simulations and real-world examples to demonstrate SDN's performance in hyper-scale solutions. The results showed that the efficiency, low latency, and high bandwidth utilization of the huge network had improved. Finally, we outline the risks and constraints, including security, compatibility, and dependability, along with potential remedies and future directions for SDN-based cloud networking research. In cloud data center environments and other relevant fields, the work provided in this paper contributes to the advancement of SDN towards high simplicity and scalability.

Lorican and Cutamora [16] examines how the centralized control and programmability of Software-Defined Networking (SDN) might enhance network security. By separating the control and data-moving components, SDN makes networks easier to manage and secure in real time. Traditional networks are frequently fixed and sluggish to react to emerging cyberthreats. Bohol Northern Star College IT specialists were surveyed and interviewed as part of the study to learn about the advantages, difficulties, and current SDN security procedures. To identify and halt assaults more quickly, a system design was developed that combines SDN with tools like firewalls, intrusion detection systems, and machine learning. The findings demonstrated how SDN enhances network monitoring, policy enforcement, and threat detection, increasing network security and adaptability. Nevertheless, the study also discovered that SDN carries hazards, including software flaws and the requirement for cautious controller configuration to prevent emerging security issues. Overall, SDN offers a powerful way to increase network security, especially in contexts like CC and IoT, if combined with strong security measures and multi-controller configurations to increase resilience.

3. PROPOSED METHODOLGY

In this article, we present a Scalable and Secure Framework for Cloud-Based IT Infrastructure Using Software-Defined Networking and Distributed Storage (SFCIT-SDNDS). The SFCIT-SDNDS technique involves data collection, and module development processes. SDN will identify any threats in the various domains that make up the network and send the encrypted data to the blockchain. Our SFCIT-SDNDS system can use real-world network data to indicate a wide range of applications, as can our new network paradigm. A set of experiment was performed to analyze the scalability of the SFCIT-SDNDS approach. Figure 1 demonstrates the working structure of the SFCIT-SDNDS technique.

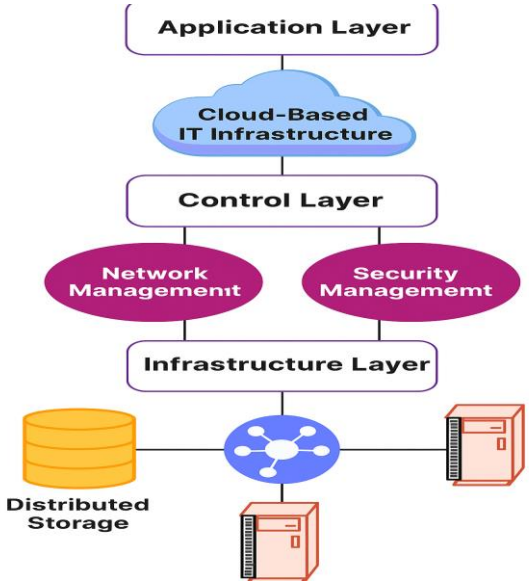

Figure 1. Working structure of the SFCIT-SDNDS approach

3.1. Data Collection

Data Collection Method A systematic framework is created to collect data regarding the security practices of SDN today, the difficulties encountered, and the advantages that are thought to exist. In-depth information about SDN security experiences and difficulties is also gathered by interviewing network administrators and security specialists. The foundation of the suggested work is the development of an CC network that satisfies the needs of smart city applications. The network builds a sophisticated, programmatically optimized CC network using SDN technology. Any physical network should be designed as simply as possible while enhancing the network's adaptability and interoperability. In general, this network's structure is separated into three levels, which include:

3.2. Data forwarding Layer

CC networks are made up of nodes that enable the network with cloud data in order to implement optimal and successful judgments. Conversely, SDN makes up virtual devices, particularly Openflow Switches, which serve as Cluster Heads (CH). This study proposed a clustering technique to efficiently arrange the CC nodes for improved network management with the resources at hand.

A cluster could have several network nodes, such as $(I_1, I_2, I_3, \dots, I_m)$. These CC nodes have the ability to sense their surroundings and gather information. CH is in charge of every cluster. Every CH now has an SDN Controller installed. Managing and controlling the cluster area while offering protection from both internal and external threats is the main goal of the SDN controller.

Gateway Nodes (GN) facilitate connectivity between clusters. CH and SDN-IoT gateways work together to manage the SDN controller, and the control plane completes the data layer routing process to send traffic via the SDN OpenFlow protocol.

3.3. Control Layer

The control layer consists of a collection of virtualized resources and multi-functional controllers. It provides guidelines for virtualized services and packet transmission behavior for applications in smart cities. In order to reduce congestion issues, the controller's tasks must be

assigned. The three primary types of network controllers used by the researchers are application, packet, and security controllers. Unauthorized network applications can be tracked using the application controller's configuration. The packet controller is in charge of load balancing and packet monitoring. The security controller adds the three extended controllers, which include the main, intrusion, and crypto controllers. These controllers, also known as high-level applications, are used to maintain honesty, anonymity, and secrecy across the network service.

3.4. Application Layer

The application fields of the constructed network are deployed in this layer, which is regarded as the topmost layer of the system architecture. Applications for smart cities, including smart houses, smart energy, smart automobiles, smart health, smart transportation, etc., are included in this layer. Furthermore, it involves cloud and server infrastructures that share content and provide real-time services to the user. This layer's primary responsibilities are data processing and service provision.

3.5. Architectural model

A user initiates network communication when they send a request to any cluster. Given a network scenario, each cluster, let's call $cluster_i$, should have a predetermined threshold value, TH_i, that indicates the maximum number of requests that this cluster can handle. A local SDN controller connects each cluster's nodes, which say I where $(I \rightarrow I_1, I_2, I_3, \ldots)$. This controller is in charge of keeping an eye on how many requests are being handled inside its own cluster at any given time. The SDN controller sends a request to another cluster via GN when the threshold value is equal to the number of requests that have already been processed. This prevents any new requests from being fulfilled in its own cluster. For processing, this request must once more be examined at the just arrived cluster. When a request is available for processing, the SDN controller lets it in and sends it to the node in its cluster based on that node's remaining availability for processing requests. Equation (1) is used to verify this at each node.

$$current - req_{I1,cluster_{i+1}} < req_{I1,cluster_{i+1}} \quad (1)$$

In this case, $req_{I1,cluster_{i+1}}$ refers to the number of requests that node I corresponding to cluster can process, and $current - req_{I1,clusier_{i+1}}$ represents the number of requests that node I is now serving in $cluster_{i+1}$. If there are several nodes that can handle this request, it can be routed to any of them by increasing the $current - req$ count by one. Equation (2) calculates the threshold value TH_{i+1}, which determines how many requests every node I in its cluster, say $cluster_{i+1}$, can process.

$$TH_{i+1} req_{I1,cluster_{i+1}} = \bar{I} \quad (2)$$

In Eq. (2), I denotes the total number of nodes in the cluster and TH_{i+1} refers to the predetermined threshold of $cluster_{i+1}$. The algorithm of scalable and secure SDN–cloud framework with distributed storage is given below.

Algorithm: Scalable and Secure SDN–Cloud Framework with Distributed Storage

Variables: req_i, $node_j$, $storage_k$, TH_sec
Procedure input: client requests (service, storage, or compute)
Begin: req_i
If $req_i \leq TH\_sec$
    forward req_i to SDN controller
    controller assigns optimum path for req_i
  else
    reject request and log security alert
  end if
 If $node_j$ is available
    assign req_i to $node_j$
    check $storage\_k$ availability
      If $storage\_k$ capacity $\geq req_i$
        store data in distributed $storage\_k$
      else
        migrate data to next available $storage\_k + 1$
      end if
  else
    reroute req_i to backup $node_{j+1}$
  end if
  notify client of request status
End

4. RESULTS AND DISCUSSION

In this section, a set of network parameters and key performance indicators (KPI) can be calculated, and an evaluation SFCIT-SDNDS design plan is created from a security standpoint. Three network topologies with different QoS measures have been constructed by the authors in this part to compute load flow and compare them using the LSTM, RF, CNN, and KNN algorithms. The entire simulation process is carried out in a Mininet emulator using OpenFlow switches, which replicate actual networking situations. Wireshark packets are captured for analysis and computation of the throughput, response time, and CPU utilization of the various topologies. Based on a number of QoS criteria, including data transfer, reaction time, and CPU utilization, the experimental result discovered that the suggested SFCIT-SDNDS algorithms outperform the current techniques in terms of efficiency.

4.1. Comparative Analysis of Data Transmission

The quantity of incoming requests received on the internet at various time intervals is known as the data transmission. Figure 2 shows that the suggested SFCIT-SDNDS algorithm achieves the highest data transmission when compared to the existing techniques. For example, when the load is 100 Mbps, the data transmission is 200 Mbps; when the load is 900 Mbps, the data transmission is 800 Mbps, and so on.

The data transmission rate is increasing in tandem with the traffic load. More data transmission results from increased cluster load. In terms of data transfer, the suggested algorithm has demonstrated improvements of 97.57%,89.25%,96.12%, and 95.45% over the LSTM, RF, CNN, and KNN algorithms, respectively.

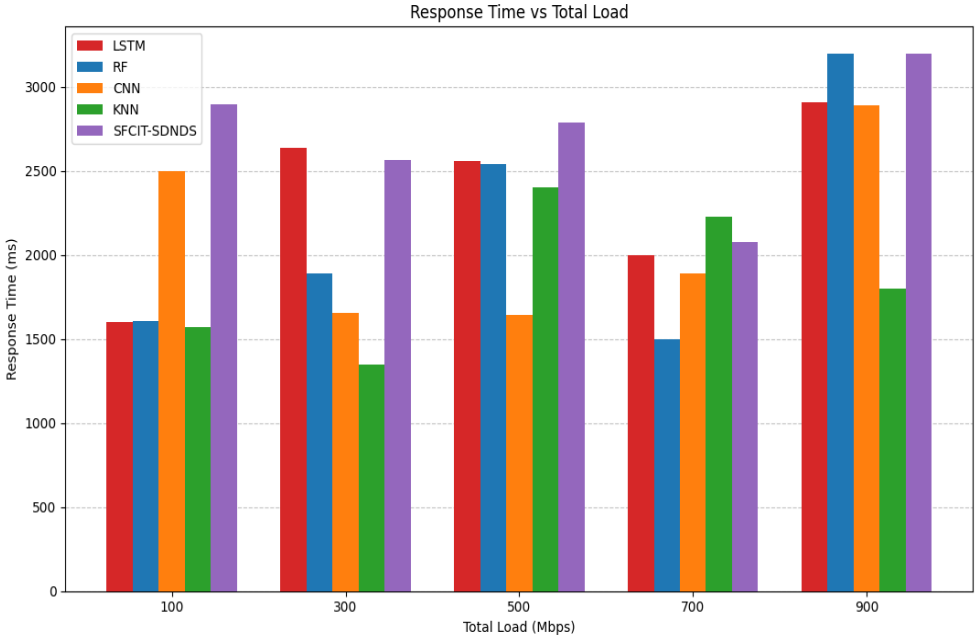

Figure 2**.** Data transmission rate of SFCIT-SDNDS with other existing models

4.2. Comparative Analysis of Average Response Time

The response time shows how many incoming requests the clusters can handle. The suggested SFCIT-SDNDS method, which has been successful in producing distinctive flows for the CC nodes, has a low response time, as shown in Figure 3.

According to the outcome, the reaction time is 0 ms for a 100 Mbps load, 10 ms for a 300 Mbps load, and so forth. In terms of response time, the suggested SFCIT-SDNDS method outperformed the LSTM, RF, CNN, and KNN by 96.24%, 97.26%, 95.36%, and 92.63%, respectively.

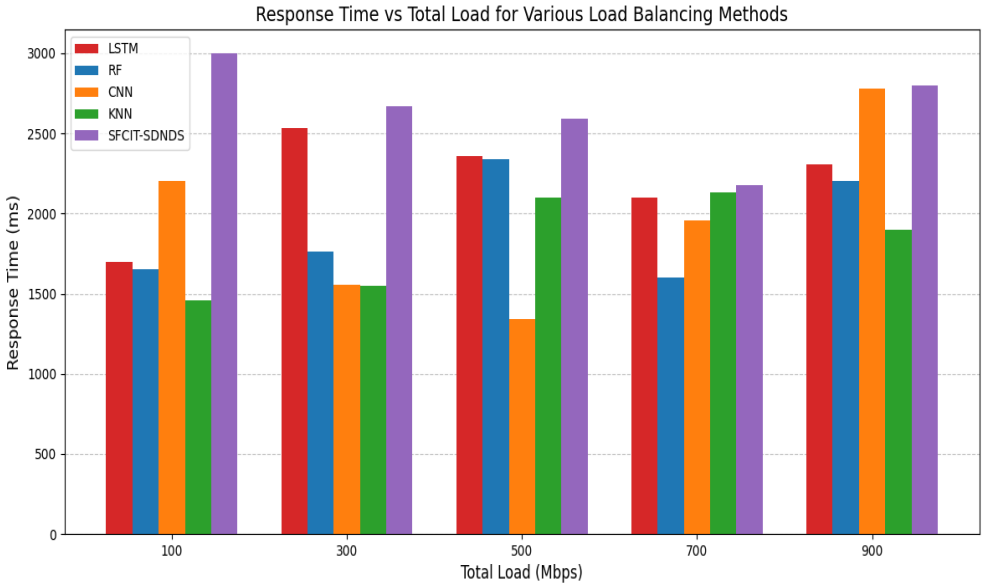

Figure 3. Response time of SFCIT-SDNDS with other existing approaches

4.3. Comparative Analysis of CPU Utilization

Various algorithms on different load on the clusters to balance the load on the various servers, i.e., servers 1, 2, and 3. As can be observed, the suggested SFCIT-SDNDS approach results in load imbalance between CC nodes if resources are used appropriately, and the CPU consumption of the existing techniques in Figure 4 is higher across all servers. In terms of CPU utilization, the suggested approach has outperformed the LSTM, RF, CNN, and KNN algorithms by 94%, 96%, 82%, and 96%, respectively.

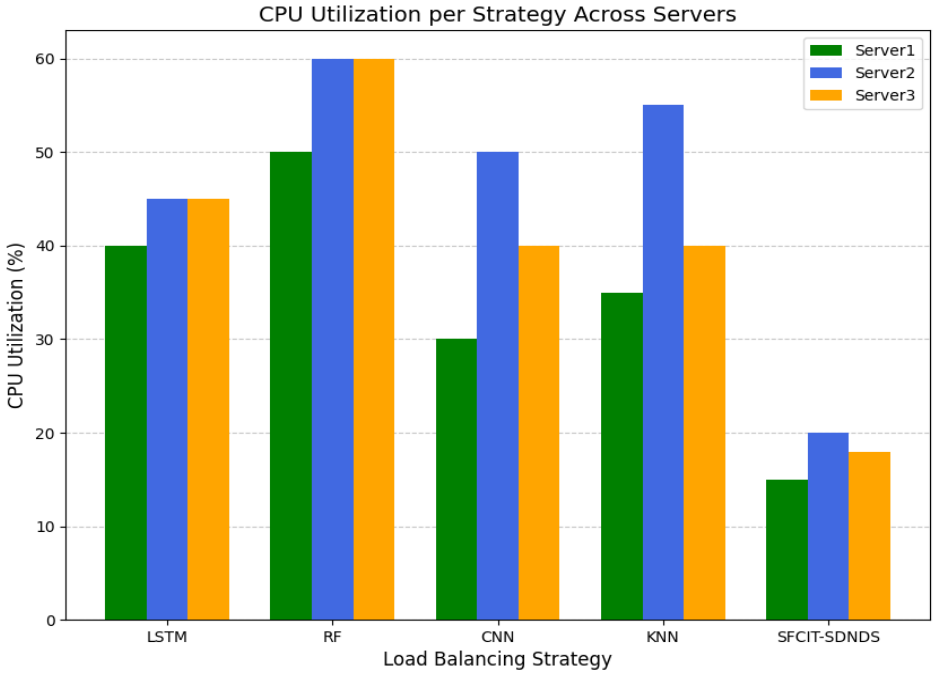

Figure 4. CPU Utilization of SFCIT-SDNDS with other existing approaches

A comparative study of several security mechanisms in SFCIT-SDNDS systems is shown in Figure 5, with an emphasis on three important factors: resource overhead, complexity, and effectiveness. Encryption, authentication, intrusion detection, resilient design, and network monitoring and anomaly detection are the five security mechanisms that are assessed in the bar chart. Each measure is scored on a % scale for each facet. Overall, effectiveness scores highly across all metrics, suggesting that they have a significant influence on SDN security. Resilient design and authentication exhibit higher levels of complexity, indicating that they call for more advanced implementation.

Additionally, resource overhead varies, with intrusion detection and resilient design having notably significant overhead due to the operational and computational expenses of these security mechanisms. By showing the balance between the resources needed for their deployment and the effectiveness of these security measures, this visualization aids in identifying the trade-offs associated with their implementation. A clear method for classifying and displaying the security risks connected to the various SFCIT-SDNDS architectural components. Every threat is grouped according to the SFCIT-SDNDS plane it impacts, and each one is described in detail to assist readers comprehend its characteristics and possible effects on the network. Understanding the vulnerabilities in an SFCIT-SDNDS architecture and how to mitigate them with suitable security solutions requires this type of aggregation.

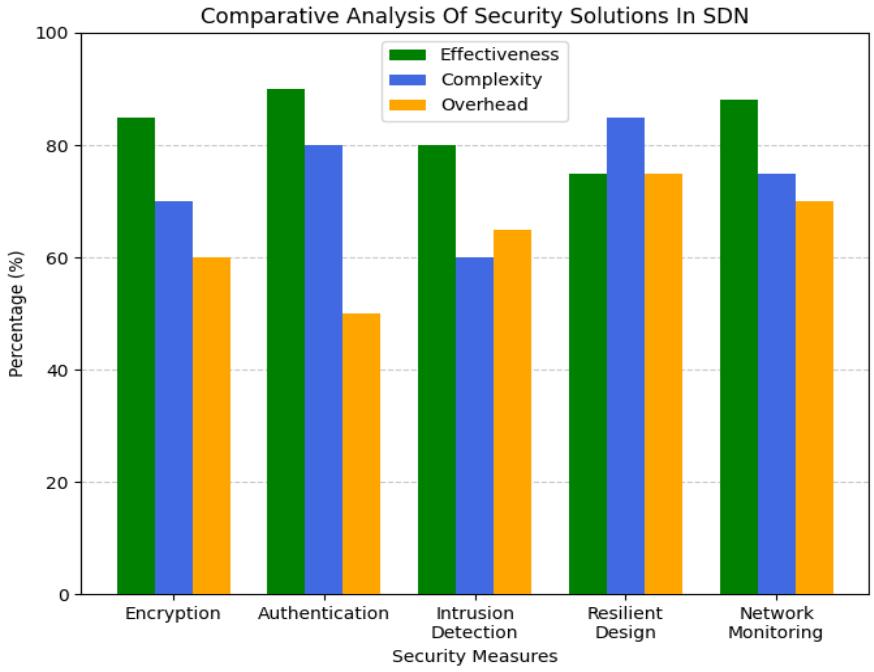

Figure 5. Comparative analysis of security solution in SFCIT-SDNDS approaches

5. CONCLUSION

In this study, we present a Scalable and Secure Framework for Cloud-Based IT Infrastructure Using Software-Defined Networking and Distributed Storage (SFCIT-SDNDS). The SFCIT-SDNDS technique involves data collection, and module development processes. SDN will identify any threats in the various domains that make up the network and send the encrypted data to the blockchain. Our

SFCIT-SDNDS system can use real-world network data to indicate a wide range of applications, as can our new network paradigm. A set of experiment was performed to analyze the scalability of the SFCIT-SDNDS model. According to our comprehensive experimental examination of performance evaluation, the suggested SFCIT-SDNDS framework outperforms other existing approaches by 97.75%.

REFERENCES

[1] Farahmandian, S., 2021. *An Interaction-based Software-Defined Security Model and Platform to secure cloud resources*. University of Technology Sydney (Australia).

[2] Krishnan, P., Jain, K., Aldweesh, A., Prabu, P. and Buyya, R., 2023. OpenStackDP: a scalable network security framework for SDN-based OpenStack cloud infrastructure. *Journal of Cloud Computing*, *12*(1), p.26.

[3] Sahito, S.F., Detho, S., Shah, S.A.R., Kim, T.H. and Noh, S.Y., A Comprehensive Research on Integration of Cloud with Software-Defined Network.

[4] Singh, A., Bali, R.S. and Aujla, G.S., 2021. Prospective on technical considerations for edge–cloud cooperation using software-defined networking. In *Software defined internet of everything* (pp. 147-176). Cham: Springer International Publishing.

[5] Sharma, R., 2021. A review on software defined networking. *Int J Sci Res Comput Sci Eng Inf Technol*, pp.11-14.

[6] Batista, E., Figueiredo, G. and Prazeres, C., 2022. Load balancing between fog and cloud in fog of things based platforms through software-defined networking. *Journal of King Saud University-Computer and Information Sciences*, *34*(9), pp.7111-7125.

[7] Awasthi, C., Sehgal, I., Pal, P.K. and Mishra, P.K., 2022. Software-defined network (SDN) for cloud-based Internet of Things. In *Transforming Management with AI, Big-Data, and IoT* (pp. 185-213). Cham: Springer International Publishing.

[8] Shafiq, Shakila, Md Sazzadur Rahman, Shamim Ahmed Shaon, Imtiaz Mahmud, and ASM Sanwar Hosen. "A Review on Software-Defined Networking for Internet of Things Inclusive of Distributed Computing, Blockchain, and Mobile Network Technology: Basics, Trends, Challenges, and Future Research Potentials." *International Journal of Distributed Sensor Networks* 2024, no. 1 (2024): 9006405.

[9] Shinde, A. and Bhingarkar, S., 2024, July. Application of soft computing techniques and software defined networks for detection of fraudulent resource consumption attacks: A comprehensive review. In *AIP Conference Proceedings* (Vol. 3028, No. 1, p. 020027). AIP Publishing LLC.

[10] Razvan, F. and Mitica, C., 2025. Enhancing network security through integration of game theory in software-defined networking framework. *International Journal of Information Security*, *24*(3), p.100.

[11] Jalali, M. and Reed, M.J., 2024, August. Securing software defined networking in cloud infrastructure through transport layer security. In *Proceedings of the 2024 8th International Conference on Cloud and Big Data Computing* (pp. 80-85).

[12] Vimal, V., Muruganantham, R., Prabha, R., Arularasan, A.N., Nandal, P., Chanthirasekaran, K. and Reddy Ranabothu, G., 2022. Enhance Software-Defined Network Security with IoT for Strengthen the Encryption of Information Access Control. *Computational Intelligence and Neuroscience*, *2022*(1), p.4437507.

[13] Van Anh, D., Chehri, A., Quy, N.M., Hue, C.T.M., Nguyen, D.C. and Quy, V.K., 2024, December. An Software Defined Networking (SDN) Enhanced Edge Computing Framework for Internet of Healthcare Things (IoHT). In *GLOBECOM 2024-2024 IEEE Global Communications Conference* (pp. 415-419). IEEE.

[14] Thangavel, S., Sunkara, K.C. and Srinivasan, S., 2022. Software-Defined Networking (SDN) in Cloud Data Centers: Optimizing Traffic Management for Hyper-Scale Infrastructure. *International Journal of Emerging Trends in Computer Science and Information Technology*, *3*(3), pp.29-42.

[15] Lorican, R.B. and Cutamora, F.J.P., 2025. Software-defined networking (SDN) for enhanced network security. *GAS Journal of Engineering and Technology (GASJET)*, *2*(4), pp.22-33.

[16] Babbar, H., Rani, S., Singh, A., Abd-Elnaby, M. and Choi, B.J., 2021. Cloud based smart city services for industrial internet of things in software-defined networking. *Sustainability*, *13*(16), p.8910.