Annals of Emerging Technologies in Computing (AETiC)
Vol. 9, No. 5, 2025

Research Article

A Lightweight Security Framework for Edge Layer IoT Networks using Neural Cryptography and Virtualization

Kavita Agrawal[1,2,*], Padala Prasad Reddy[1] and Suresh Chittineni[3]

[1]Andhra University, Vishakhapatnam, India
kavita.courses@gmail.com; prasadreddy.vizag@gmail.com
[2]Chaitanya Bharathi Institute of Technology, Hyderabad, India
kavitaagrawal_cet@cbit.ac.in
[3]GITAM University, Vishakhapatnam, India
schittin@gitam.edu
*Correspondance: kavita.courses@gmail.com

Received: 19 January 2025; Accepted: 15 July 2025; Published: 25 October 2025

Abstract: Securing the edge layer is essential in modern cybersecurity architectures, particularly for the Internet of Things (IoT), where resource-constrained devices require robust yet lightweight protection mechanisms. This paper introduces a novel Neural Cryptography Secure Router (NCSR) framework that integrates Tree Parity Machine (TPM)-based neural key generation with AES encryption, OpenWRT-based firewalling, and a virtualized intrusion detection/prevention system. The architecture is implemented using Raspberry Pi devices at the edge and a Fedora-based host for virtualization and centralized security processing. The framework features two Raspberry Pi units: the first simulates an IoT node, encrypting sensor data with TPM-generated keys before transmission via SSH/SCP; the second operates as a secure router, running OpenWRT and nftables for real-time packet filtering. The Fedora host functions as a multi-layered security hub, hosting virtual machines (pfSense and Security Onion) for firewalling, deep packet inspection, and threat analysis via Snort and Suricata. This integrated model eliminates the need for pre-shared keys while ensuring end-to-end confidentiality and dynamic session key exchange. Empirical evaluations demonstrate strong performance with minimal resource consumption: 1.2 ms/KB encryption time, 1.1 ms/KB decryption time, 25% CPU utilization, 95.5% firewall drop efficiency, and a 7% false positive rate. Comparative analysis with existing solutions confirms the model's advantages in terms of security, scalability, and computational efficiency, establishing NCSR as a practical and novel security solution for IoT edge networks.

Keywords: ***AES Encryption; Edge Layer Security; Firewall; Intrusion Detection and Prevention System (IDS/IPS); Neural Cryptography***

1. Introduction

The Internet of Things (IoT) has emerged as one of the most innovative paradigms of the digital era. It interconnects billions of devices across healthcare, industry, and urban infrastructures. In spite of its benefits, IoT networks remain exposed to a wide variety of threats as data travels through heterogeneous layers and resource-constrained devices. Security in such environments is particularly challenging due to limited computational power, memory, and energy resources. A comprehensive survey of IoT threats emphasizes that while blockchain, fog computing, and machine learning are often proposed as countermeasures, no single approach provides a holistic, resource-aware solution for IoT protection [1].

Kavita Agrawal, Padala Prasad Reddy and Suresh Chittineni, "A Lightweight Security Framework for Edge Layer IoT Networks using Neural Cryptography and Virtualization", *Annals of Emerging Technologies in Computing (AETiC)*, Print ISSN: 2516-0281, Online ISSN: 2516-029X, pp. 43-60, Vol. 9, No. 5, 25 October 2025, Published by International Association for Educators and Researchers (IAER), DOI: 10.33166/AETiC.2025.05.004, Available: http://aetic.theiaer.org/archive/v9/v9n5/p4.html.

The growth of the Social IoT (SIoT) paradigm, integrating IoT with social networking principles, adds complexity to the security landscape. The inherent heterogeneity of devices, lack of standardized protocols, and increasing dependency on trust ecosystems open new attack surfaces, mainly at the perception, transportation, and application layers [2]. Similarly, broader reviews of IoT evolution demonstrate that although technical and legal safeguards exist, the absence of robust end-to-end frameworks continues to hinder widespread and secure IoT adoption [3].

Among emerging approaches, neural cryptography has emerged as a lightweight, adaptive solution for secure communication. Early studies showed that two artificial neural networks, when trained on exchanged outputs, could synchronize weights and generate a shared secret key over a public channel [4]. Complementary IoT-driven models for smart city applications demonstrate how integrating predictive analytics, real-time data processing, and robust protocols can enhance responsiveness while maintaining data confidentiality and resilience [5].

Lightweight cryptographic protocols have also been extensively studied as a means to mitigate the mismatch between strong security demands and device limitations. Modular lightweight solutions, incorporating distributed key renewal mechanisms and scalable architectures, achieve data confidentiality and integrity while introducing negligible overhead [6].

Virtualization and edge-oriented designs provide complementary advantages in addressing IoT security. Frameworks such as EdgeMeld illustrate how distributed architectures at the edge can integrate anomaly detection and adaptive machine learning to mitigate latency, improve scalability, and enhance robustness against attacks [7]. Similarly, adversarial neural cryptography using generative adversarial networks (GANs) has been explored to autonomously establish secure connections in IoT systems without human intervention, reinforcing the adaptability of cryptographic defenses [8].

Adversarial robustness has also gained importance in the IoT security domain. Deep learning-based systems, while powerful, are highly vulnerable to adversarial perturbations [9]. Likewise, generative adversarial defense mechanisms have been applied to strengthen continuous-variable quantum key distribution systems, eliminating adversarial perturbations and securing sensitive communications [10].

Further advances in neural-based cryptosystems highlight the potential of pseudo-random number generators and non-linear neural encryption schemes to enhance confidentiality in IoT environments [11]. Complementary work on lightweight cryptographic protocols tailored for constrained IoT devices demonstrates reductions in time and space complexity while sustaining high levels of security, validating their practicality for real-world deployments [12].

In response to these gaps, this study proposes a novel security framework tailored for edge-layer IoT networks. This paper aims to create an end-to-end Security Framework for IoT networks with a secure router that manages traffic on the network, employs Neural Cryptography for encrypting information, and utilizes virtualization and a Root of Trust at the local IoT hub to facilitate process isolation and detect harmful components.

This framework aims to maintain minimal computational overhead while delivering robust, real-time security for IoT and small-scale edge networks. The proposed system targets applications ranging from localized IoT deployments to long-distance secure communications.

2. Literature Survey

The increase in IoT devices and edge computing has led to significant research in lightweight cryptographic techniques, real-time intrusion detection, and adaptive network security models. Below are key studies that have influenced the development of the proposed system.

Dall *et al*. [13] introduced KVM/ARM, a hypervisor integrated into the Linux kernel, which exploits ARM's Hyp mode with minimal code changes. Their work showed that virtualization could be lightweight and feasible even for ARM-based devices, overcoming hardware challenges. Inspired by this, the proposed work incorporates virtualization techniques (using lightweight hypervisors) to securely isolate critical components of the system, particularly on the Fedora Hub platform. While

promising, most hypervisor-based solutions incur non-negligible computational overhead, highlighting the challenge of balancing security isolation with real-time IoT performance.

Kinzel *et al.* [14] proposed neural cryptography using Tree Parity Machines (TPM) for dynamic key generation, offering faster synchronization and lower hardware costs compared to traditional cryptography. Despite vulnerabilities highlighted by Klimov *et al.* [15], who exposed potential attack vectors, the concept remains attractive for resource-constrained IoT environments. Building on this, Meraouche *et al.* [16] proposed a multi-agent adversarial neural network model where Alice and Bob autonomously learn to use asymmetric public/private keys to secure communication against eavesdroppers. Their study extended neural cryptography beyond symmetric setups, showing resilience against stronger adversaries such as leakage and chosen-plaintext attacks. Sun *et al.* [19] proposed a Neural Cryptographic Framework for secure key generation and distribution, leveraging lightweight CNN-based encryption. Collectively, these works illustrate how neural cryptography evolved from early TPM-based symmetric models to more adaptive adversarial and deep learning frameworks. However, most remain validated only in simulated or small-scale IoT environments, and comprehensive real-world testing is still lacking. These pioneering ideas motivate the integration of neural cryptography in our proposed work for secure key generation and encrypted data transfer between IoT devices and the centralized hub.

Langiu *et al.* [17] proposed UpKit's modular update framework that emphasizes secure firmware distribution across heterogeneous IoT platforms, reinforcing the importance of device-agnostic, resilient systems. Similarly, Moratelli *et al.* [18] stressed the critical role of embedded virtualization in creating isolated execution domains, ensuring that even compromised processes cannot affect the entire device. These findings inspired the isolation of system components through virtualization while maintaining efficient, update-friendly architectures at the edge and hub layers. Errabelly *et al.* [23] also presented a micro-hypervisor model to isolate crucial processes, reducing malware risks. Taken together, these studies confirm virtualization's role in IoT resilience but reveal a common gap: they prioritize isolation while often neglecting lightweight cryptographic mechanisms, leaving communication channels less protected. Through a fusion of neural learning, embedded virtualization, and secure routing, the paper proposes a novel system-level security framework for modern IoT and edge computing environments.

Using Virtual Network Security Functions (VNSFs), such as VPNs and Intrusion Prevention Systems (IPS), Canavese *et al.* [20] presented a security virtualization framework. Although their strategy emphasizes network-layer security, device-level protection is not entirely covered. This concept is used in the proposed work by combining pfSense firewall and Security Onion (IDS/IPS) for network-layer and edge-device security. An adaptive edge-based intrusion detection system that uses machine learning for anomaly detection was proposed by Hagan *et al.* [21]. According to their research, hybrid IDS models increase security but need to be optimized for devices with limited resources. The current system extends this idea by integrating Security Onion IDS/IPS in the Fedora Hub and using a Raspberry Pi 2 for local traffic filtering. Tiburski *et al.* [22] investigated pfSense's function in network-layer security, showing how well it works for traffic segmentation and DDoS mitigation. Comparing these approaches reveals that while firewalls and IDS solutions enhance network defense, they often operate independently from key management protocols, resulting in fragmented security that our integrated model seeks to address.

Although they pointed out the significant computational overhead, Errabelly *et al.* [23] demonstrated how micro-hypervisors can confine malware to isolated domains. Khan *et al.* [24] investigated blockchain-based trust mechanisms for safe IoT communication. While blockchain enhances data integrity, its energy demands make it unsuitable for lightweight IoT devices. This contrast highlights the trade-off between strong trust guarantees and resource efficiency. Our system addresses these concerns through neural cryptography, which provides confidentiality and trust without excessive computational cost.

Dai *et al.* [25] proposed an edge-driven security framework for intelligent IoT systems, emphasizing the integration of edge and cloud computing to expand service capabilities while

identifying critical adversarial threats. Höglund *et al.* [26] introduced BLEND, a framework that combines secure communication and storage at the application layer while keeping latency low. Ehui *et al.* [27] proposed a lightweight mutual authentication protocol for IoT nodes and gateways using symmetric-key cryptography and frequent session key updates. Both works highlight the need to reduce overhead in IoT security. Together, these works show that while lightweight authentication and storage frameworks are emerging, a unified design that couples efficient encryption, intrusion detection, and virtualization is still missing. Our framework follows this direction by providing real-time, efficient encryption and secure router-based traffic management.

Other studies have focused on making IoT security scalable and future-ready. Ashrif *et al.* [28] proposed PSLAE, a provably secure and lightweight authenticated encryption protocol for machine-to-machine communication in IIoT, addressing challenges such as denial-of-service, forward secrecy, and ephemeral secret leakage while keeping storage and computation costs low. Their findings demonstrated that secure and efficient communication can be achieved in constrained environments through lightweight encryption. Rehman *et al.* [29] introduced QESIF, a quantum-enhanced IoT framework that integrates quantum key distribution with intrusion detection to ensure resilience and low latency in smart city deployments. These studies collectively highlight the tension between near-term lightweight cryptographic solutions (e.g., PSLAE) and long-term quantum-ready frameworks (e.g., QESIF). This dual perspective reinforces the need for adaptable security frameworks that can evolve with future technological advances.

The proposed work synthesizes advancements from multiple research domains to create a comprehensive, secure IoT framework. It adopts neural cryptography for lightweight, dynamic key generation, inspired by Sun *et al.*, while integrating virtualized IDS/IPS and pfSense firewall as explored by Canavese *et al.* and Tiburski *et al.* The system incorporates edge-based anomaly detection and virtualization strategies from Hagan *et al.* and Errabelly *et al.*, ensuring security at both the network and device levels. By leveraging Raspberry Pi devices for traffic filtering and encrypted data transmission, the project delivers a cost-effective, scalable, and resource-efficient security model for modern IoT environments. Table 1 encapsulates recent scholarly endeavors concerning IoT and edge security, emphasizing cryptographic techniques, virtualization, and trust mechanisms.

Table 1. Comparative analysis of recent approaches in IoT and edge security frameworks

Paper	Approach	Strengths	Limitations
Dall *et al.* [13]	Kernel-based Virtual Machine on ARM (KVM/ARM) lightweight hypervisor	Demonstrated feasibility of virtualization on ARM devices	Kernel-level integration only; limited scalability
Kinzel *et al.* [14]	Neural cryptography using Tree Parity Machines (TPM)	Low hardware cost; fast synchronization	Vulnerable to known attack vectors
Klimov *et al.* [15]	Security analysis of Tree Parity Machines (TPM)	Identified attack surfaces in neural cryptography	Exposed weaknesses in basic TPM design
Meraouche *et al.* [16]	Adversarial neural cryptography (multi-agent model)	Public/private key learning; resilient against leakage attacks	Computational overhead for constrained IoT devices
Langiu *et al.* [17]	UpKit secure update framework	Modular; device-agnostic firmware updates	Requires customization for heterogeneous platforms
Moratelli *et al.* [18]	Embedded virtualization	Strong domain isolation	Lacks network-layer security integration
Sun *et al.* [19]	Neural cryptography for IoT authentication using Convolutional Neural Networks (CNN)	Lightweight CNN-based encryption; efficient session key distribution	Limited real-world validation
Canavese *et al.* [20]	Security virtualization using Virtual Network Security Functions (VNSFs)	Modular and scalable; enhances network-layer security	Ignores device-level constraints
Hagan *et al.* [21]	Edge-based intrusion detection using hybrid Machine Learning Intrusion Detection Systems (ML-IDS)	Detects anomalies dynamically; adaptable filtering	Resource heavy for IoT devices
Tiburski *et al.* [22]	pfSense (open-source firewall) for IoT	Robust segmentation; Distributed Denial of Service (DDoS) mitigation	Network-level only; no cryptographic support

Errabelly *et al.* [23]	Micro-hypervisor for malware isolation	Process isolation; supports edge security	High computational overhead
Khan *et al.* [24]	Blockchain-based trust management for IoT	Ensures distributed trust and integrity	Energy-expensive; poor for lightweight IoT
Dai *et al.* [25]	Edge-driven IoT security framework	Integration of edge and cloud for resilience	Complex management
Höglund *et al.* [26]	BLEND (secure communication and storage framework)	Secure communication and storage; low latency	Focused on application-layer only
Ehui *et al.* [27]	Lightweight mutual authentication protocol	Frequent key updates; simple design	Limited scalability studies
Ashrif *et al.* [28]	Provably Secure and Lightweight Authenticated Encryption (PSLAE)	Provably secure; efficient for Machine-to-Machine (M2M) communication	Constrained to Industrial IoT (IIoT)
Rehman *et al.* [29]	Quantum-Enhanced Security for IoT Framework (QESIF) using Quantum Key Distribution (QKD) and Intrusion Detection System (IDS)	Combines QKD with IDS for futureproofing	Requires quantum infrastructure

Table 1 highlights that existing IoT security frameworks often focus on isolated aspects such as encryption, virtualization, or trust management, but rarely offer an integrated solution. Neural cryptography-based methods [19, 21] are lightweight but lack system-level validation. Virtualization approaches [20, 22] ensure isolation but omit cryptographic protection. Others [23–24] address scalability and adaptability but fall short on lightweight encryption. In contrast, the proposed NCSR framework integrates neural key generation, AES encryption, firewalling, and virtualized IDS/IPS into a cohesive, low-overhead system, effectively bridging these gaps and enabling scalable, real-world IoT security.

3. Proposed Design

The proposed system implements a multi-layered IoT security framework combining neural cryptography, AES encryption, firewalling, and virtualized IDS/IPS. As shown in Figure 1, sensor data from Raspberry Pi Dev 1 is encrypted using a TPM-based neural key and AES-EAX, then transmitted securely via SSH. Raspberry Pi Dev 2, running OpenWRT, filters network traffic and logs potential threats using Snort and NAT. The Fedora-based IoT hub receives the encrypted data, verifies session keys via Diffie–Hellman exchange, and processes it through virtual machines (pfSense and Security Onion) for deep packet inspection and threat analysis.

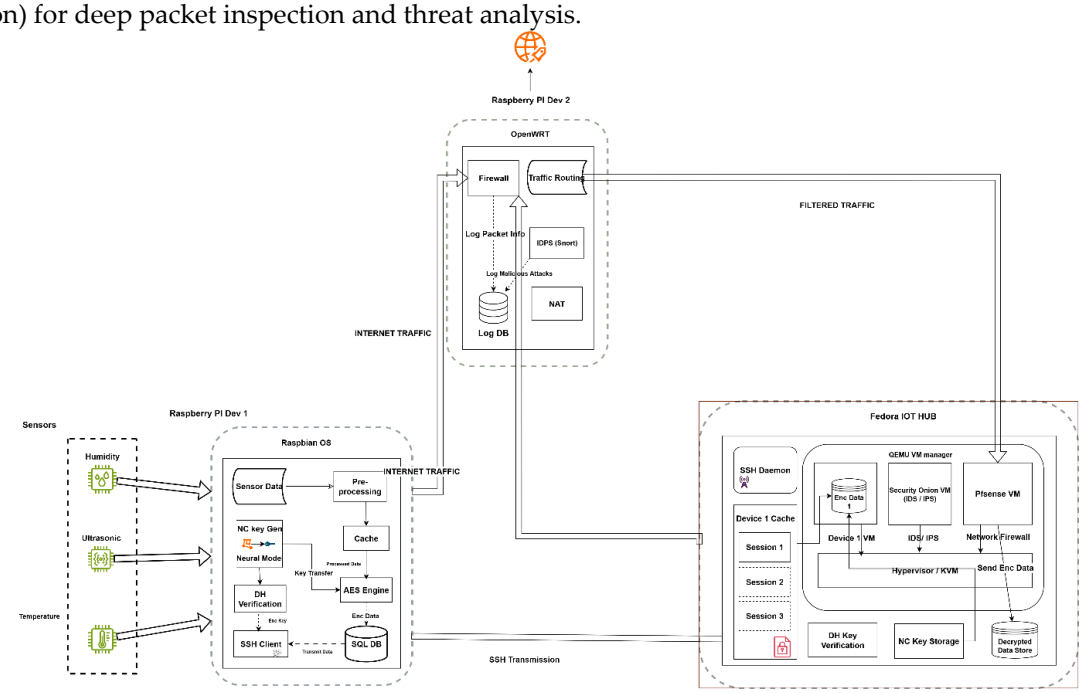

Figure 1. End-to-End Edge-Layer Security Architecture Using Neural Cryptography and Virtualized Intrusion Detection in IoT

The IOT network (Figure 2) consists of various sensors (Humidity, Temperature, Ultrasonic) connected to a microcomputer (Raspberry Pi 1), A second Raspberry Pi running OpenWRT is configured as a secure wireless access point which provides internet connectivity to IOT Hub running Fedora OS which collects data from various devices.

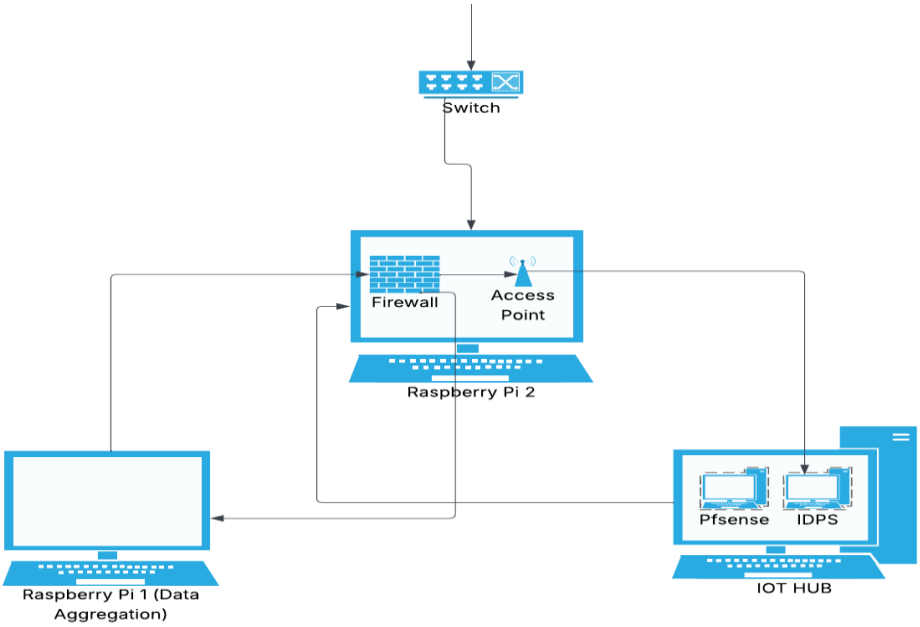

Figure 2. Network Structure of the Secure IoT Edge-to-Hub Communication Framework

3.1 Main Components:

The following are the components (Figure 1) in detail:

3.1.1. Data Collection Node (Raspberry Pi Dev 1)

- Sensors: Collect real-time environmental data (e.g., Humidity, Ultrasonic, Temperature).
- Sensor Data & Preprocessing: Raw sensor data is captured and optionally to remove noise or irrelevant metrics.
- Neural Cryptography (NC) Key Generation: Generates session keys using neural networks for enhanced cryptographic strength.
- DH Key Verification: Verifies exchanged keys using the Diffie-Hellman method before encryption.
- AES Engine: Performs AES encryption on the processed sensor data.
- SQL Database: Encrypted data is temporarily stored locally.
- SSH Client: Securely transmits encrypted data to the central IoT Hub via SSH.

3.1.2. OpenWRT Router (Raspberry Pi Dev 2)

- Firewall & Traffic Routing: Filters and routes both internet traffic and internal communications.
- IDPS Integration: A lightweight Intrusion Detection and Prevention System monitors traffic and logs malicious activities and anomalies to a log database.
- NAT (Network Address Translation): Assists in proper packet forwarding and IP masking within the local network.

3.1.3. Fedora IoT Hub (Centralized Processing Server)

- SSH Daemon: Receives encrypted data from edge devices.
- Device Cache: Temporarily stores session data for each connected device.
- OEMU Virtual Machine Manager: Hosts different VMs for modular functions:
- Security Onion VM: Intrusion Detection and Packet Inspection.
- PfSense VM: Manages network firewall rules and inspection.
- DH Key Verification & Neural Key Storage: Verifies session keys before decryption.

- Decryption Engine: Decrypts received data using AES and validated keys.
- Decrypted Data Store: Saves the final readable data for further analysis, visualization, or archival.

3.2. Network Structure

The network structure of the proposed secure IoT communication framework is depicted in Figure 2. It is composed of distributed edge security nodes and a centralized processing hub, each serving critical roles in ensuring end-to-end data confidentiality and integrity. Raspberry Pi 1 and Raspberry Pi 2 are deployed at the edge level, functioning as security-enhanced nodes. Raspberry Pi 1 is primarily responsible for data aggregation, while Raspberry Pi 2 operates as a traffic-filtering node with firewall and access point capabilities. Both devices collaboratively capture, pre-process, and filter network traffic before securely relaying the information to the central unit.

The Fedora-based IoT Hub acts as the core of the security architecture. It handles the management of firewall policies, monitors intrusion detection system (IDS) alerts, and hosts containerized or virtualized security services such as pfSense and Security Onion. This modular virtualization approach enhances the flexibility and scalability of the system in dynamic threat environments. To safeguard data in transit, secure SSH channels are established between the Raspberry Pi devices and the Fedora Hub. These channels facilitate encrypted communication, thereby mitigating risks associated with man-in-the-middle (MITM) attacks and data interception.

3.3 Proposed Algorithm

Algorithm 1. System-Level Secure Framework using Neural Cryptography and Virtualization

Component 1: Raspberry Pi 1 – Data Collection and Encryption

Initialize sensors and local SQLite database
Train and sync TPM-based Neural Cryptography model
while device is active **do**
  Read temperature, humidity sensors
  Store readings in local database with timestamp
  if neural key not yet synchronized **then**
  Begin TPM synchronization with Hub
  if successful **then**
  Save session key
  else
  Retry
  end if
  end if
  for all stored rows **do**
  Encrypt data using AES-EAX with neural key
  Insert ciphertext and nonce into encrypted database
  end for
  Archive encrypted DB as .tar.gz
  Transmit archive via SCP to Fedora Host
  if Transmission success **then**
  Clear plaintext and encrypted entries
  end if
  Wait for next cycle
end while

Component 2: Raspberry Pi 2 – Secure Routing & Firewall (OpenWRT)

Initialize OpenWRT router; enable DHCP, DNS, NAT
Configure firewall rules using UCI to filter IPs/domains
Enable traffic logging via logread, tcpdump
while router is active **do**
  Monitor packets and apply firewall filters
  if Intrusion or policy match **then**
  Drop or log traffic
  end if
end while

Component 3: Fedora Host – Decryption and Virtual Security Hub

34: Start SSH daemon; listen for incoming SCP uploads
35: **while** new archive received **do**
36: Decompress and extract archive
37: Re-sync TPM model to generate neural session key
38: **for all** cipher rows in archive **do**
39: Decrypt using AES-EAX with neural key
40: Store data in centralized SQLite database
41: **end for**
42: Forward decrypted data to cloud backup
43: **end while**

Virtualization Layer (Fedora Hub)

44: Start QEMU/KVM VMs:
- pfSense VM for firewall and routing analysis
- Security Onion VM for IDS/IPS traffic monitoring

45: Connect virtual bridge for traffic flow via VMs
46: Monitor alerts and logs for anomalies using Security Onion

Raspberry Pi 1 serves as the data collection and preliminary encryption unit. It initializes the connected sensors and maintains a local SQLite database to store real-time readings. A Tree Parity Machine (TPM)-based Neural Cryptography model is trained and synchronized with the Fedora Hub for secure session key generation. Sensor data is collected, encrypted using AES-EAX mode with the neural session key, and then stored in an encrypted database. The database is archived and securely transmitted to the Fedora Host via SCP. After successful transmission, plaintext and encrypted entries are cleared. Raspberry Pi 2 functions as a secure router and firewall node using OpenWRT. It provides DHCP, DNS, and NAT services and enforces firewall rules configured through Unified Configuration Interface (UCI). Packet monitoring is enabled, and traffic matching intrusion or policy violations is either dropped or logged.

The Fedora Host acts as the central decryption and security processing unit. It listens for incoming SCP uploads, decompresses and extracts archives, and re-synchronizes the TPM model to regenerate the neural session key. Received cipher entries are decrypted using AES-EAX and stored in a centralized SQLite database. The decrypted data is then forwarded to cloud backup services.The virtualization environment is deployed on the Fedora Hub to enhance security and modularity. QEMU/KVM virtual machines are launched, hosting pfSense for firewall and routing analysis, and Security Onion for intrusion detection and monitoring. A virtual bridge connects the network interfaces for proper traffic flow and centralized security monitoring.

4. Implementation

4.1. Hardware and Software Requirements

The proposed framework is implemented using lightweight and cost-effective hardware, supported by a modular software stack optimized for secure data collection, encryption, transmission, and monitoring. The specifications are summarized in Tables 2 and 3.

4.1.1. Hardware Specifications

The hardware setup includes two Raspberry Pi devices for edge operations and one x86_64 Fedora-based host system for centralized processing and virtualization. Additional components such as SD cards, Ethernet cables, and optional storage support flexible deployment.

Table 2. Hardware Requirements for Secure IoT Communication Architecture

Component	Specification / Purpose
Raspberry Pi 1 (Dev 1)	Model: Raspberry Pi 3B/4B Role: Sensor data collection, neural key generation, AES encryption
Raspberry Pi 2 (Dev 2)	Model: Raspberry Pi 3B/4B Role: OpenWRT-based secure router with firewall and IDPS
Fedora Host System	Processor: x86_64, RAM ≥ 8 GB Role: Central security hub with decryption and VM hosting

SD Cards (2x)	Capacity: ≥ 16 GB each Purpose: OS and file system for Raspberry Pi devices
Ethernet Cables	Wired connectivity for Pi 2 and Fedora host
Wi-Fi Module/Dongle	Enables Pi 2 to operate in Access Point (AP) mode
External Storage (Opt.)	USB/SSD for backing up decrypted data locally or to the cloud

4.1.2. Software Stack

The software components were selected to support secure data acquisition, neural key generation, virtualization, and intrusion monitoring, forming a layered and modular software ecosystem.

Table 3. Software Requirements

Software/Tool	Purpose
Raspberry Pi OS	OS for Pi 1 to support Python-based sensor interaction and encryption
OpenWRT	Lightweight router OS deployed on Pi 2 for NAT, firewall, and traffic routing
Python 3.x	Implementation of AES encryption, neural key logic, and data pre-processing
TensorFlow / Keras	Training and synchronization of Tree Parity Machine models for neural key generation
SQLite3	Embedded database used for sensor data storage at the edge and decryption output at the hub
OpenSSH / SCP	Secure remote communication and file transmission between edge nodes and the Fedora Host
Fedora Linux	OS for the host system supporting QEMU/KVM-based virtualization
QEMU / Virt-Manager	VM hosting environment for pfSense and Security Onion
Security Onion	Real-time intrusion detection and deep packet inspection system
pfSense	Network segmentation and firewall management via a virtualized interface
Wireshark (Optional)	A deep packet inspection tool is used during testing and performance verification
tar, gzip	Data compression tools are used for the secure packaging of sensor data archives

4.2. Methodology: Secure IoT Data Communication

4.2.1. Raspberry Pi 1: Data Simulation, Encryption, and Transmission

Raspberry Pi 1 is configured to emulate an environmental monitoring node by generating synthetic temperature and humidity readings within predefined ranges. These values represent the plaintext input for the encryption process and are produced in a controlled Python virtual environment to ensure consistency and reproducibility during testing (Figure 3).

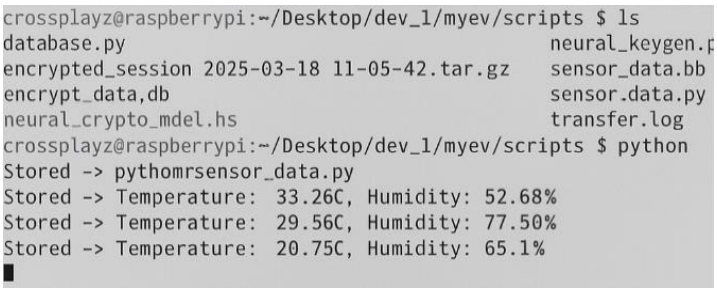

Figure 3. Generation of simulated temperature and humidity sensor data on Raspberry Pi 1

The generated sensor data is stored in a lightweight SQLite database. Each record is timestamped and uniquely indexed, and a binary state flag (0 or 1) is appended to indicate whether the record has been successfully transmitted. This approach supports fault-tolerant transmission and simplifies record tracking (Figure 4).

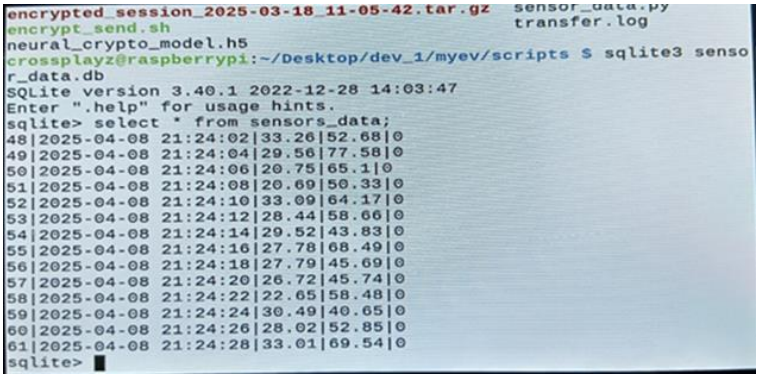

Figure 4. Storage of sensor data with timestamp and transmission status in SQLite database

Once the data is collected, it is encrypted using the AES algorithm in EAX mode. The session key is dynamically generated using a Tree Parity Machine (TPM)-based neural cryptographic model. This method enables high-entropy, lightweight key generation suitable for resource-constrained edge devices. The encrypted values are written back to the SQLite database for transmission (Figure 5)

Figure 5. Encryption of stored sensor data using a TPM-derived AES-EAX session key

Encrypted data is securely transferred to the central hub via SCP over SSH. The transmission process is automated through shell scripts, and the data is compressed into .tar.gz archives to optimize bandwidth usage. After a successful transfer, records are flagged accordingly and old data is deleted to manage storage efficiently (Figure 6).

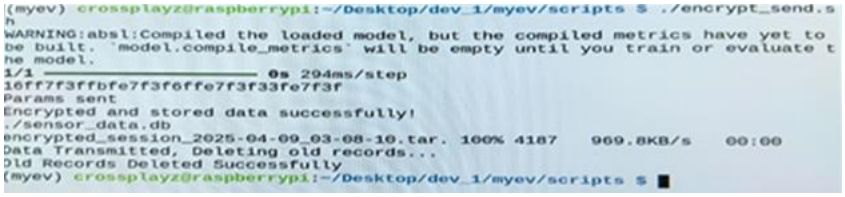

Figure 6. Secure transmission of encrypted data archive to the Fedora-based hub and deletion of old records

4.2.2. Fedora Host: Key Verification, Decryption, and Database Reconstruction

The Fedora host receives the encrypted .tar.gz archive through an SSH daemon and stores it in a session-specific directory. The shared neural session key is verified using a Diffie-Hellman (DH) key exchange protocol to ensure mutual authentication between nodes (Figure 7).

Figure 7. Key verification on the hub using Diffie-Hellman exchange for TPM session authentication

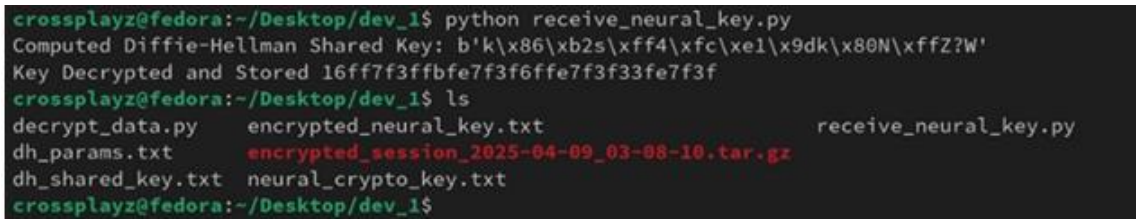

Figure 8. Received archive decompressed and prepared for decryption and analysis

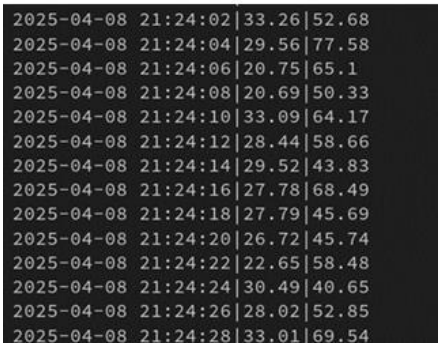

Figure 9. Successfully decrypted sensor data stored in the hub's centralized SQLite database

After successful verification, the encrypted archive is decompressed into a structured session folder. File integrity and timestamps are preserved during extraction, enabling orderly batch decryption (Figure 8).

Using the verified session key, the hub decrypts the sensor data using AES-EAX and reconstructs it into a new SQLite database. This centralized database mirrors the schema used at the edge and facilitates further analysis, storage, or visualization (Figure 9).

4.2.3. Raspberry Pi 2: OpenWRT-Based Router

OpenWRT Boot and LAN Verification

Raspberry Pi 2 is configured as a secure edge router using the OpenWRT operating system. Upon successful flashing and boot basic network connectivity is verified using ping tests to public servers (e.g., Google), confirming WAN access.

Packet Tracing and Logging

OpenWRT is equipped with logging and packet inspection tools such as logread and tcpdump. These tools allow tracing incoming and outgoing traffic, enabling detailed inspection of headers, IP addresses, and protocols. This configuration supports rudimentary intrusion detection capabilities

Domain Blocking via UCI Firewall

The firewall configuration in OpenWRT utilizes the Unified Configuration Interface (UCI) to block specific domains. IP addresses are resolved using traceroute, and custom firewall rules are appended via terminal commands. Results are validated through ping and log analysis

4.2.4. Virtual Machines at the Fedora Hub

The Fedora-based host functions as the central security hub, hosting multiple virtual machines (VMs) for modularized network security tasks. Virtualization is implemented using QEMU/KVM, managed through Virt-Manager, enabling the deployment of pfSense for firewall operations and Security Onion for intrusion detection and packet inspection. To ensure strong isolation and performance consistency, several virtualization techniques were applied. CPU pinning was configured to dedicate specific processor cores to each VM, reducing contention and improving real-time responsiveness. Virtual bridge networking was used to route traffic between VMs and physical interfaces, allowing granular monitoring and filtering. IOMMU (Input-Output Memory Management Unit) support was enabled to allow device pass-through, which enhances VM access to hardware-level interfaces while maintaining isolation. Additionally, network segmentation between VMs was enforced to minimize inter-VM traffic leakage and contain potential threats within their respective domains.

This setup ensures that each security function—firewalling, intrusion detection, and decryption—is compartmentalized, reducing the attack surface and improving system scalability. The use of virtualization also allows for independent updates and monitoring, providing operational flexibility without compromising performance or security.

4.3 Deployment Challenges and Considerations

Real-world IoT environments often exhibit non-deterministic network latency, heterogeneous hardware performance, and administrative overhead due to distributed security components. These factors may hinder consistent operation and increase system maintenance complexity. In response, the proposed framework was designed with modularity and automation in mind:

- Latency Mitigation: By leveraging asynchronous encrypted transmission and secure session queues, the system accommodates delay variance in SSH/SCP transfers.
- Hardware Heterogeneity: The use of platform-independent software (Python, OpenWRT, and KVM/QEMU) ensures compatibility across diverse edge devices.
- Maintenance Optimization: Automated key management using TPM-based neural cryptography reduces manual provisioning. System logs, IDS alerts, and VM performance are centrally managed via Kibana dashboards, improving visibility and manageability.

These design decisions enhance the framework's resilience and scalability in real-world deployments involving diverse IoT and edge infrastructure.

5. Result

This section presents the performance evaluation of the proposed security framework across cryptographic, firewall, and virtualization layers. Key metrics such as encryption time, intrusion detection efficiency, resource utilization, and system scalability are discussed.

5.1. Evaluation of Cryptographic Algorithms

To evaluate the cryptographic algorithms, the following metrics were considered:

- Encryption Time and Decryption Time (ms/KB)
- Latency
- Accuracy (% Correctly Encrypted & Decrypted)

Custom Python scripts using time.perf_counter() was utilised to measure algorithm runtimes, using:

$$Encryption\ time = End\ time - Start\ time \quad (1)$$

The accuracy was calculated using the following formula:

$$Accuracy\ (\%) = \frac{Correctly\ Decrypted\ Data}{Total\ Data\ Points} * 100 \quad (2)$$

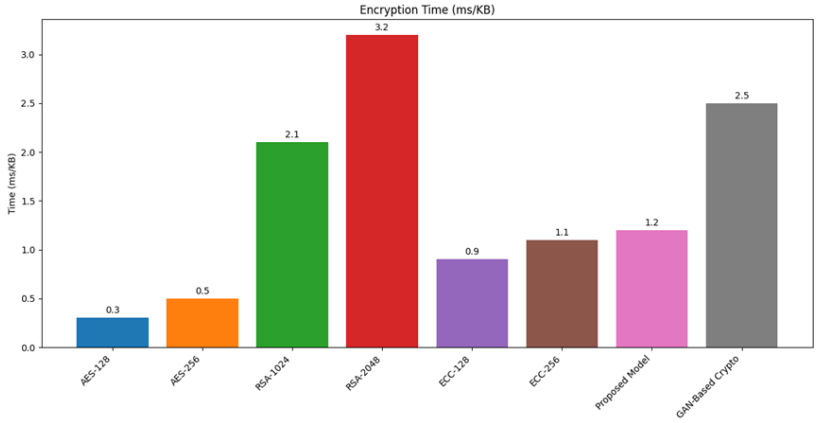

Figure 10. The proposed model achieves encryption times comparable to AES and ECC and significantly outperforms RSA and GAN-based encryption

As shown in Figure 10, the proposed model achieves a competitive encryption time compared to AES and ECC-based schemes, while outperforming RSA and GAN-based cryptography.

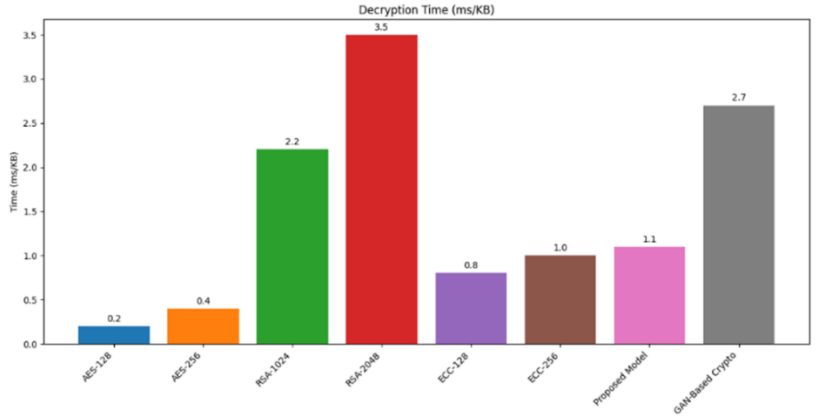

Figure 11. Comparison of decryption time (ms/KB) between standard cryptographic algorithms and the proposed model

Figure 11 illustrates the decryption time performance (in milliseconds per kilobyte) for a range of cryptographic algorithms. As observed, symmetric ciphers like AES-128 achieve the lowest decryption

time, whereas asymmetric schemes like RSA-2048 incur the highest. The proposed model achieves a decryption time of 1.1 ms/KB, positioning itself favorably against existing methods.

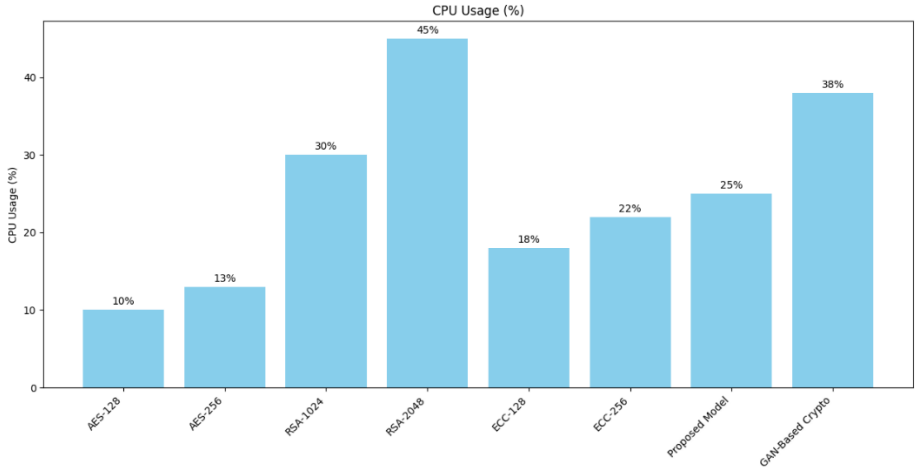

Figure 12. CPU efficiency of various encryption Algorithms

As illustrated, the proposed approach demonstrates moderate CPU utilization (25%), significantly lower than RSA (45%) and GAN-based models (38%), supporting its applicability in constrained IoT devices.

5.2 Comparative Evaluation with Existing IoT Security Frameworks

To contextualize the performance and design of the proposed Neural Cryptography Secure Router (NCSR), a comparative evaluation is presented in Table 4. This comparison includes recent state-of-the-art lightweight IoT security frameworks, selected based on relevance to resource-constrained environments, cryptographic efficiency, and modular design. The table highlights differences in encryption latency, CPU utilization, scalability, and key management strategies.

Table 4. Comparative Evaluation of Lightweight IoT Security Frameworks

Framework	Encryption Latency (ms/KB)	CPU Utilization (%)	Modularity	Scalability	Key Management
Proposed NCSR (this work)	~1.2	~25	High (TPM + virtualized stack)	Excellent (up to 100 nodes)	TPM-based neural key + AES (no PSK)
BLEND [26]	~0.11 /packet (~0.63 ms/KB)	Low (minimal CPU)	Medium (application-layer only)	Moderate	PKI-based key management
Decision-Tree IDS [27]	<1 ms (real-time inference)	Very low (~<1 ms runtime)	High (edge + cloud integration)	High	Automated updates; model-managed keys
SPiME (PiM-based AES) [28]	~0.04 (ms/KB equivalent)	Very low (<5%)	Low (hardware-specific)	Very high (4000+ units)	AES-128 key encapsulation
QESIF (Quantum-enhanced IoT) [29]	~20 ms average latency	Low energy per session	Medium (hybrid stack)	High	QKD + classical hybrid key exchange

This analysis demonstrates that the proposed NCSR framework strikes a strong balance between computational efficiency and layered security architecture. Unlike frameworks relying solely on pre-shared keys or hardware acceleration, NCSR leverages neural cryptography with AES-EAX in a modular virtualized environment, making it both secure and adaptable to edge IoT deployments.

5.3. Security Analysis of Neural Cryptography

Tree Parity Machine-based neural cryptography provides lightweight and adaptive key generation. Unlike RSA and ECC, TPM relies on neural synchronization rather than hard mathematical problems. However, potential vulnerabilities such as synchronization and man-in-the-middle (MITM) attacks necessitate additional safeguards. The framework addresses this by integrating Diffie-Hellman key verification and employing AES-EAX for authenticated encryption.

Table 5. Comparison of TPM-Based Neural Cryptography with RSA and ECC

Feature / Metric	TPM-Based Neural Crypto	RSA	ECC
Key Generation Approach	Neural synchronization (TPM)	Prime factorization	Elliptic curve log
Key Entropy	High	High	High
Computational Cost	Low	High	Moderate
Suitability for IoT Devices	Excellent	Poor	Fair
Resistance to MITM Attacks	Moderate (with DH verification)	Strong	Strong
Encryption Strength	Enhanced via AES-EAX	Strong	Strong
Vulnerability	Synchronization attacks	Quantum threats	Quantum threats
Cryptographic Maturity	Emerging	Established	Established

By combining TPM with AES-EAX and DH verification, the framework achieves both lightweight performance and improved resistance to cryptographic attacks, making it suitable for secure IoT communication.

5.4. Firewall on OpenWRT router (Raspberry Pi 2)

To evaluate the effectiveness of the firewall and intrusion detection system (IDS) deployed on Raspberry Pi 2 running OpenWRT, several performance metrics were analyzed. The key parameters include firewall rule evaluation time, packet drop efficiency, false positive rate, and traffic throughput.

To evaluate the firewall effectiveness, the following metrics were considered:

- Firewall Rule Evaluation Time
- Packet Drop Efficiency (%)
- False Positive Rate (%)
- Traffic Throughput (Mbps)

The *tcpdump* was employed to capture and analyze real-time traffic logs; and *hping3* was used to simulate DDoS attacks and generate malformed packets to test the system's resilience. The following evaluation formulas were used:

$$Drop\ Efficiency\ (\%) = \frac{(Dropped\ Malicious\ Packets)}{(Total\ Malicious\ Packets)} * 100 \quad (3)$$

$$False\ Positive\ Rate = \left(\frac{Benigh\ Packet\ Flagged}{Total\ benign\ Packets}\right) * 100 \quad (4)$$

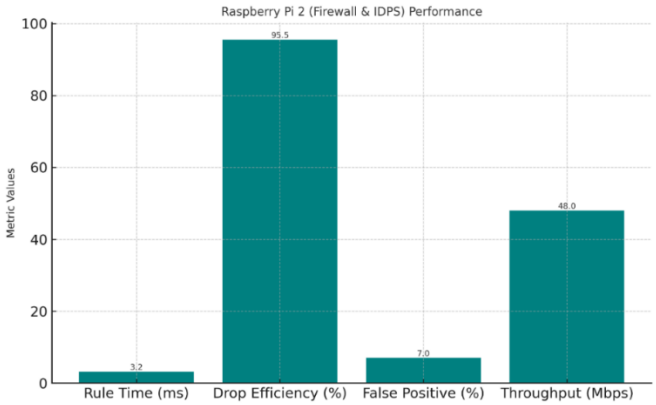

Figure 13. Firewall and IDPS Performance

The firewall was configured using UCI to define custom rules for domain and IP filtering. During testing, the system effectively detected and blocked malicious traffic while maintaining stable throughput. As shown in Figure 13, the configuration achieved a packet drop efficiency of 95.5% and sustained a throughput of 48 Mbps. The false positive rate remained within an acceptable range at 7%, demonstrating reliable filtering without excessive disruption to legitimate traffic. These results confirm that the OpenWRT-based Raspberry Pi 2 router is capable of handling edge-layer security operations efficiently, even under simulated attack conditions.

5.5. Virtualization Performance on Fedora Host

To evaluate the virtualization performance, the following metrics were considered:

- VM Boot Time
- CPU Isolation Efficiency
- Memory Overhead
- Event Logging Rate (events/sec)

The tools used include: htop, virt-top, dstat for resource monitoring and Security Onion dashboard (via Kibana) for IDS analysis. The following evaluation formulas were used:

$$\text{Event Logging Rate} = \frac{Total\ Logged\ Events}{Observation\ time\ (s)} \quad (4)$$

$$\text{CPU Isolation Efficiency} = \left(1 - \frac{CPU\ Utilization\ Leakage}{Total\ CPU\ Assigned}\right) * 100 \quad (5)$$

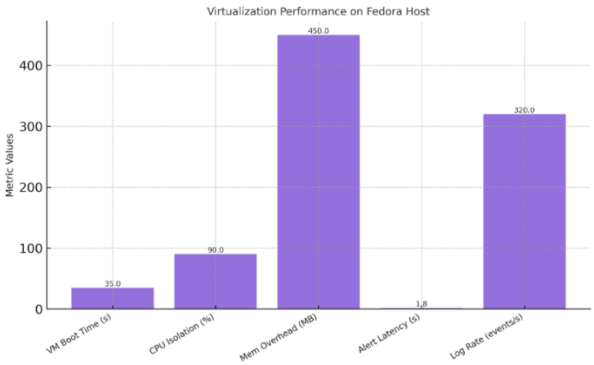

Figure 14. Virtualization performance on the Host

As depicted in Figure 14, the Fedora host maintained a memory overhead of approximately 450 MB and a logging rate of 320 events/sec, confirming the virtualization layer's ability to handle real-time analysis with minimal performance degradation.

Table 6. Secure IoT Framework Evaluation Summary

Component	Evaluation Metric	Relative Performance
Cryptographic Engine	Encryption/Decryption Speed	Moderate — Balanced for real-time IoT usage
	Key Generation Adaptability	High — Dynamic per session using neural crypto
	Security Strength	Strong — Combines AES with neural key logic
	Computational Overhead	Low — Optimized for edge devices
Raspberry Pi 2 (Firewall & IDPS)	Packet Filtering Efficiency	High — Effectively drops unauthorized packets
	Intrusion Detection Capability	Moderate — Detects common and abnormal patterns
	Processing Load	Low — Suitable for lightweight deployments
	False Alarm Rate	Moderate — Tuned with custom rules
Fedora Host (Virtualization Layer)	VM Isolation and Resource Control	High — Secure and efficient via KVM/QEMU
	Scalability of Security Infrastructure	High — Supports expansion and monitoring
	Monitoring and Logging Speed	Efficient — Near real-time alert generation
Key Management & Exchange	Session Key Flexibility	High — Unique keys for each transmission
	Secure Exchange Protocol	Robust — Utilizes Diffie-Hellman securely
	Synchronization Delay	Low — Acceptable for real-time use
Overall System Integration	Modularity and Component Interoperability	High — Each module functions independently
	Suitability for IoT/Edge Networks	Excellent — Designed for constrained devices
	Maintenance and Update Feasibility	Manageable — Uses standard tools and scripts

The results in Table 6 demonstrate that the proposed system achieves high adaptability, strong security, and low overhead across all layers, making it well-suited for real-time IoT and edge network deployments.

5.6. Scalability Analysis

To evaluate the scalability of the proposed framework, simulated workloads were generated to emulate deployments of 10, 25, 50, and 100 IoT nodes transmitting encrypted data to the virtualized security hub. Each node was represented by a synthetic traffic stream mimicking the frequency and volume of real-world sensor communications. Tools such as tcpreplay and iperf3 were used to simulate

traffic load, while system-level resource usage was monitored using virt-top, htop, and dstat. The table below summarizes key performance metrics measured on the Fedora host, including average CPU utilization, memory overhead, and sustained network **throughput**.

Table 7. Resource Utilization Under Varying IoT Node Loads

Number of IoT Nodes	Average CPU Usage (%)	Memory Overhead (MB)	Throughput (Mbps)
10	18	320	50
25	26	370	72
50	34	410	94
100	45	470	116

As shown in table 7, the system demonstrates linear growth in CPU and memory consumption as the number of connected nodes increases. Importantly, throughput scales consistently, and no performance bottlenecks were observed up to 100 simulated nodes. These results indicate that the framework can accommodate larger IoT deployments without significant degradation in performance, confirming its scalability for real-time edge-layer applications.

6. Conclusion

This work presents a comprehensive and lightweight framework for securing IoT edge communications through the integration of TPM-based neural cryptography, AES encryption, firewall, and virtualized IDS/IPS systems. The proposed NCSR architecture demonstrates a novel combination of neural key exchange, virtual machine-based segmentation, and rule-based intrusion detection tailored to the needs of constrained IoT devices. Its modular structure and efficient performance make it highly suitable for scalable, real-time deployments in edge computing environments. Experimental results confirm the system's practical viability, achieving competitive encryption and decryption times, high packet filtering efficiency, and low resource overhead. The framework also addresses the key management challenge in IoT systems by eliminating the need for static pre-shared keys and enabling dynamic key generation. Future enhancements to the proposed framework will focus on increasing its adaptability, intelligence, and cryptographic resilience. First, the integration of AI-driven intrusion detection mechanisms—such as long short-term memory (LSTM) networks and autoencoders—will be explored to improve anomaly detection, particularly for identifying zero-day attacks and dynamic threat patterns. Second, federated learning will be investigated as a means to enable distributed IDS/IPS training across multiple edge devices without centralizing sensitive data, thereby enhancing both privacy and responsiveness. Finally, the cryptographic layer will be extended to evaluate post-quantum encryption algorithms, ensuring long-term security in the face of quantum computing advancements. These developments aim to establish a more intelligent, privacy-preserving, and future-proof edge security framework.

CRediT Author Contribution Statement

Kavita Agrawal: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing; Padala Prasad Reddy: Conceptualization, Methodology, Supervision, Validation, Writing – review & editing; Suresh Chittineni: Resources, Project administration, Supervision, Writing – review & editing.

References

[1] Naqash Azeem Khan, Azlan Awang and Samsul Ariffin Abdul Karim, "Security in Internet of Things: A review", *IEEE Access*, Online ISSN: 2169-3536, Vol. 10, 26 July 2022, pp. 104649–104670, Published by IEEE, DOI: 10.1109/ACCESS.2022.3209355, Available: https://ieeexplore.ieee.org/document/9902998.

[2] Mario Frustaci, Pasquale Pace, Gianluca Aloi and Giancarlo Fortino, "Evaluating critical security issues of the IoT world: Present and future challenges", *IEEE Internet of Things Journal*, Online ISSN: 2327-4662, Vol. 5, No. 4, 1 August 2018, pp. 2483–2495, Published by IEEE, DOI: 10.1109/JIOT.2017.2767291, Available: https://ieeexplore.ieee.org/document/8086136.

[3] Oludare Isaac Abiodun, Esther Omolara Abiodun, Moatsum Alawida, Rami S. Alkhawaldeh and Humaira Arshad, "A review on the security of the internet of things: Challenges and solutions", *Wireless Personal*

Communications, E-ISSN: 1572-834X, Print ISSN:0929-6212, Vol. 119, 2021, pp. 2603–2637, Published by Springer, DOI: 10.1007/s11277-021-08348-9, Available: https://link.springer.com/article/10.1007/s11277-021-08348-9.

[4] Wolfgang Kinzel and Ido Kanter, "Neural cryptography", in *Proceedings of the 9th International Conference on Neural Information Processing (ICONIP)*, 18-22 November 2002, Singapore, Print ISBN: 981-04-7524-1, Vol. 3, pp. 1351–1354, Published by IEEE, DOI: 10.1109/ICONIP.2002.1202841, Available: https://ieeexplore.ieee.org/document/1202841.

[5] Kalyankumar Dasari, Mohmad Ahmed Ali, Shankara N. B., K. Deepthi Reddy, M. Bhavsingh *et al.*,"A Novel IoT-Driven Model for Real-Time Urban Wildlife Health and Safety Monitoring in Smart Cities", in *Proceedings of the 8th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 3-5 October 2024, Kirtipur, Nepal, E-ISBN:979-8-3503-7642-5, pp. 122–129, Published by IEEE, DOI: 10.1109/I-SMAC61858.2024.10714601, Available: https://ieeexplore.ieee.org/document/10714601.

[6] Pedro Rosa, André Souto and José Cecílio, "Light-SAE: A lightweight authentication protocol for large-scale IoT environments made with constrained devices", *IEEE Transactions on Network and Service Management*, E-ISSN: 1932-4537, Vol. 20, No. 3, pp. 2428-2441, September 2023, Published by IEEE, DOI: 10.1109/TNSM.2023.3275011, Available: https://ieeexplore.ieee.org/document/10122630.

[7] K. Lakshmi, Garlapadu Jayanthi, and Jallu Hima Bindu, "EdgeMeld: An Adaptive Machine Learning Framework for Real-Time Anomaly Detection and Optimization in Industrial IoT Networks", *International Journal of Computer Engineering in Research Trends*, Print ISSN: 2349-0829, Vol. 11, No. 4, 1 April 2024, pp. 20–31, Published by IJCERT, DOI: 10.22362/ijcert/2024/v11/i4/v11i403, Available: https://www.ijcert.org/index.php/ijcert/article/view/951.

[8] Basil Hanafi and Mohammad Ubaidullah Bokhari, "Enhancement of security in connection establishment for IoT infrastructure through adversarial neural cryptography using GANs", in *Proceedings of the 3rd International Conference on ICT for Digital, Smart, and Sustainable Development (ICIDSSD 2022)*, 24-25 March 2022, New Delhi, India, E-ISBN: 978-1-63190-396-0, E-ISSN: 2593-7642, Published by European Alliance for Innovation (EAI). DOI: 10.4108/eai.24-3-2022.2318924, Available: https://eudl.eu/doi/10.4108/eai.24-3-2022.2318924.

[9] Weimin Zhao, Qusay H. Mahmoud and Sanaa Alwidian, "Evaluation of GAN-based model for adversarial training", *Sensors*, Online ISSN: 1424-8220, Vol. 23, No. 5, March 2023, Published by MDPI, DOI: 10.3390/s23052697, Available: https://www.mdpi.com/1424-8220/23/5/2697.

[10] Xun Tang, Pengzhi Yin, Zehao Zhou and Duan Huang, "Adversarial perturbation elimination with GAN-based defense in continuous-variable quantum key distribution systems", *Electronics*, Online ISSN: 2079-9292, Vol. 12, No. 11, 2023, Art. No. 2437, Published by MDPI, DOI: 10.3390/electronics12112437, Available: https://www.mdpi.com/2079-9292/12/11/2437.

[11] Apdullah Yayık and Yakup Kutlu, "Neural Network-Based Cryptography", *Neural Network World*, Print ISSN: 1210-0552, Vol. 24, No. 2, 1st April 2014, pp. 177–192, Published by Institute of Computer Science AS CR, Prague., DOI: 10.14311/NNW.2014.24.011, Available: https://nnw.cz/doi/2014/NNW.2014.24.011.pdf.

[12] Plabon Bhandari Abhi, Kristelle Ann R. Torres, Tao Yusoff and K. Samunnisa, "A Novel Lightweight Cryptographic Protocol for Securing IoT Devices", *International Journal of Computer Engineering in Research Trends*, Print ISSN: 2349-0829, Vol. 10, No. 10, 15th October 2023, pp. 24–30, Published by IJCERT, DOI: 10.22362/ijcert/2023/v10/i10/v10i104, Available: https://www.ijcert.org/index.php/ijcert/article/view/875.

[13] Christoffer Dall and Jason Nieh, "KVM/ARM: The Design and Implementation of the Linux ARM Hypervisor", *ACM SIGPLAN Notices*, Online ISSN: 0362-1340, Vol. 49, No. 4, 1 April 2014, pp. 333–348, Published by ACM, DOI: 10.1145/2644865.2541946, Available: https://dl.acm.org/doi/10.1145/2644865.2541946.

[14] Wolfgang Kinzel and Ido Kanter, "Neural Cryptography", in *Proceedings of the 9th International Conference on Neural Information Processing (ICONIP)*, 18-22 November 2002, Singapore, Print ISBN: 981-04-7524-1, Vol. 3, pp. 1351–1354, Published by IEEE, DOI: 10.1109/ICONIP.2002.1202841, Available: https://ieeexplore.ieee.org/document/1202841.

[15] Alexander Klimov, Anton Mityagin, and Adi Shamir, "Analysis of Neural Cryptography", in *Advances in Cryptology—ASIACRYPT 2002, Lecture Notes in Computer Science*, Online ISBN: 978-3-540-36178-7, Print ISBN: 978-3-540-00171-3, Series Print ISSN: 0302-9743, Series Online ISSN: 1611-3349, Vol. 2501, pp. 288–298, DOI: 10.1007/3-540-36178-2, Published by Springer Berlin, Heidelberg, Available: https://link.springer.com/chapter/10.1007/3-540-36178-2_18.

[16] Ishak Meraouche, Sabyasachi Dutta , Haowen Tan and Kouichi Sakurai", Learning asymmetric encryption using adversarial neural networks", *Engineering Applications of Artificial Intelligence*, Online ISSN: 1873-6769, Print ISSN: 0952-1976, Vol. 123, August 2023, 106220, DOI: 10.1016/j.engappai.2023.106220, Available: https://www.sciencedirect.com/science/article/abs/pii/S0952197623004049.

[17] Alessio Langiu, Carlo Alberto Boano, Matthias Schuß and Kay Römer, "UpKit: An Open-Source, Portable, and Lightweight Update Framework for Constrained IoT Devices", in *Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, 7–10 July 2019, Dallas, TX, USA, E-ISBN:978-1-7281-2519-0, pp. 2101–2112, Published by IEEE, DOI: 10.1109/ICDCS.2019.00207, Available: https://ieeexplore.ieee.org/document/8884933.

[18] Carlos Moratelli, Sergio Johann, Marcelo Neves and Fabiano Hessel, "Embedded Virtualization for the Design of Secure IoT Applications", in *Proceedings of the 27th International Symposium on Rapid System Prototyping (RSP)*, 06-07 October 2016, Pittsburgh, PA, USA, E-ISBN:978-1-4503-4535-4, pp. 1–5, Published by IEEE, DOI: 10.1145/2990299.2990301, Available: https://ieeexplore.ieee.org/document/7909116.

[19] Yingnan Sun, Frank P.-W. Lo and Benny Lo, "Lightweight Internet of Things Device Authentication, Encryption, and Key Distribution Using End-to-End Neural Cryptosystems", *IEEE Internet of Things Journal*, Online ISSN: 2327-4662, Vol. 9, No. 16, 15 August 2022, pp. 14978–14990, Published by IEEE, DOI: 10.1109/JIOT.2021.3067036, Available: https://ieeexplore.ieee.org/document/9381407.

[20] Daniele Canavese, Luca Mannella, Leonardo Regano and Cataldo Basile, "Security at the Edge for Resource-Limited IoT Devices", *Sensors*, Online ISSN: 1424-8220, Vol. 24, No. 2, 8 January 2024, Article No. 590, Published by MDPI, DOI: 10.3390/s24020590, Available: https://www.mdpi.com/1424-8220/24/2/590.

[21] Matthew Hagan, Fahad Siddiqui and Sakir Sezer, "Enhancing Security and Privacy of Next-Generation Edge Computing Technologies", in *Proceedings of the 17th International Conference on Privacy, Security and Trust (PST)*, 26-28 August 2019, Fredericton, NB, Canada, E-ISBN:978-1-7281-3265-5, Published by IEEE, DOI: 10.1109/PST47121.2019.8949052, Available: https://ieeexplore.ieee.org/abstract/document/8949052.

[22] Abou Bakary Ballo and Diarra Mamadou, "A Comprehensive Study of IoT Security Issues and Protocols", *International Journal of Computer Engineering in Research Trends*, Print ISSN: 2349-0829, Vol. 10, No. 7, 15 July 2023, pp. 8–14, Published by IJCERT, DOI: 10.22362/ijcert/2023/v10/i07/v10i0702, Available: https://www.ijcert.org/index.php/ijcert/article/view/858.

[23] Kewei Sha, Ranadheer Errabelly, Wei Wei, T. Andrew Yang and Zhiwei Wang "EdgeSec: Design of an Edge Layer Security Service to Enhance IoT Security", *Proceedings of the IEEE/ACM 1st International Conference on Fog and Edge Computing (ICFEC)*, 14–15 May 2017, Madrid, Spain, E-ISBN:978-1-5090-3047-7, Published by IEEE, DOI: 10.1109/ICFEC.2017.7, Available: https://ieeexplore.ieee.org/document/8014363.

[24] Mohd Khan, Mohsen Hatami, Wenfeng Zhao and Yu Chen, "A Novel Trusted Hardware-Based Scalable Security Framework for IoT Edge Devices", *Discover Internet of Things*, Online ISSN: 2731-4441, Vol. 4, No. 4, 2024, Published by Springer, DOI: 10.1007/s43926-024-00056-7, Available: https://link.springer.com/article/10.1007/s43926-024-00056-7.

[25] Minghui Dai, Zhou Su, Ruidong Li, Yuntao Wang, Jianbing Ni and Dongfeng Fang, "An Edge-Driven Security Framework for Intelligent Internet of Things", *IEEE Network*, Print ISSN: 0890-8044, Online ISSN: 1558-156X, Vol. 34, No. 5, September 2020, pp. 39–45, Published by IEEE, DOI: 10.1109/MNET.011.2000068, Available: https://ieeexplore.ieee.org/document/9199790.

[26] Joel Höglund and Shahid Raza, "BLEND: Efficient and Blended IoT Data Storage and Communication with Application Layer Security", in *Proceedings of the 2022 IEEE International Conference on Cyber Security and Resilience (CSR)*, 27–29 July 2022, Rhodes, Greece, E-ISBN:978-1-6654-9952-1 ,pp. 253–260, Published by IEEE, DOI: 10.1109/CSR54599.2022.9850290, Available: https://ieeexplore.ieee.org/document/9850290.

[27] Brou Bernard Ehui, Yiran Han, Hua Guo and Jianwei Liu ,"A Lightweight Mutual Authentication Protocol for IoT", *Journal of Communications and Information Networks*, Print ISSN: 2096-1081, Online ISSN: 2096-109X, Vol. 7, No. 2, June 2022, pp. 181–191, Published by China Communications Society and IEEE, DOI: 10.23919/JCIN.2022.9815201, Available: https://ieeexplore.ieee.org/document/9815201.

[28] Fatma Foad Ashrif, Elankovan A. Sundararajan, Mohammad Kamrul Hasan, Rami Ahmad, Aisha-Hassan Abdalla Hashim et. al, "Provably secured and lightweight authenticated encryption protocol in machine-to-machine communication in industry 4.0", *Computer Communications*, Online ISSN: 1873- 703X, Print ISSN: 0140-3664, Vol. 218, 15 March 2024, pp. 263-275, DOI: 10.1016/j.comcom.2024.02.008, Available: https://www.sciencedirect.com/science/article/abs/pii/S0140366424000586.

[29] Abdul Rehman and Omar Alharbi, "QESIF: A Lightweight Quantum-Enhanced IoT Security Framework for Smart Cities", *Smart Cities*, Online ISSN: 2624-6511, Vol. 8, No. 4, 1st July 2025, Article No. 116, Published by MDPI, DOI: 10.3390/smartcities8040116, Available: https://www.mdpi.com/2624-6511/8/4/116.