International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Designing Confidential Cloud Computing for Multi-Dimensional Threats and Safeguarding Data Security in a Robust Framework

Vasavi Bande[1], B. Deevena Raju[2], Karu Prasada Rao[3], Suneet Joshi[4], Sonia H. Bajaj[5], V. Sarala[6]

Submitted: 03/11/2023 **Revised**: 22/12/2023 **Accepted**: 04/01/2024

Abstract - In the dynamic landscape of cloud computing, robust security is imperative to safeguard sensitive data from cyber threats. Protecting against unauthorized access and ensuring data integrity are fundamental fostering trust and reliability in cloud services. Cyber-attacks on clouds often start with tricks like phishing or spreading harmful software. Weak passwords, mistakes in settings, or outdated systems make it easy for hackers. Once they get in, they may steal data or harm shared resources. They try to gain more control and cause damage. Stopping these attacks needs good defences and always watching for anything suspicious. Confidential computing emerges as a paramount paradigm in cloud security, establishing secure enclaves that process sensitive data within isolated, encrypted spaces. This innovative approach significantly mitigates the risk of unauthorized access, providing heightened data confidentiality beyond conventional security measures. Notably, even cloud service providers are barred from accessing data within these secure enclaves, fortifying defences against insider threats. The architecture enables the secure processing of encrypted data, maintaining encryption during usage and offering an additional layer of protection. This proves invaluable in scenarios requiring the analysis or processing of sensitive information, effectively reducing the attack surface for potential threats.

Keyword: *Cyber Attack, Confidential Computing, Cloud Security, Cloud Service Provider, Encrypted Data*

1. INTRODUCTION

Cloud computing has revolutionized the digital landscape, providing scalable and flexible solutions for handling vast volumes of data [1]. In this era of unprecedented digital growth, cloud platforms manage enormous datasets, ranging from terabytes to exabytes, enabling businesses to seamlessly store, process, and analyse information [2]. However, this surge in data utilization comes with the looming threat of cyber-attacks.

With the increasing frequency and sophistication of cyber threats, ensuring robust data security in the cloud has become paramount. Cybersecurity breaches, ranging from unauthorized access to data leaks, pose significant risks to organizations and individuals alike [3]. As businesses and individuals entrust their sensitive information to cloud services, the imperative for stringent data security measures becomes indispensable. Safeguarding against cyber threats is not merely a technological challenge, it's a fundamental necessity to uphold trust, confidentiality, and the integrity of data in the digital age.

In the ever-evolving landscape of cybersecurity, the traditional reliance on basic encryption methods and conventional firewalls for cloud security has become insufficient. As noted in recent literature [4], these foundational measures are struggling to keep pace with the rapid evolution of cyber threats. Recognizing this, the contemporary approach to cloud security necessitates a multifaceted strategy that goes beyond the conventional. Today, advanced security measures have become imperative to safeguard sensitive data in the cloud effectively. Multi-factor authentication stands out as a critical layer of defense, adding an extra barrier against unauthorized access. Furthermore, anomaly detection mechanisms play a pivotal role in identifying and responding to unusual patterns of behavior that may signify a potential threat [5].

Encryption, a fundamental element of safeguarding data, extends its purpose beyond ensuring confidentiality alone [6]. It now extends to ensuring unintelligibility, making it a formidable defense against sophisticated cyber adversaries. Robust access controls and authentication mechanisms further contribute to a layered security approach, ensuring that only authorized individuals can access sensitive information. To keep data safe in the cloud, it's crucial to actively watch for potential threats and use real-time

[1]*Associate Professor, Department of Information Technology, MVSR Engineering College, Hyderabad, Telangana 501510 Email vasavi.bande@gmail.com.*
[2]*Senior Assistant Professor , Department of Data Science and Artificial Intelligence, Faculty of Science and Technology,, ICFAI Foundation for Higher Education, Hyderabad-501203. Email : deevenaraju@ifheindia.org.*
[3]*Assistant Professor, Department of Computer Science & Engineering, GITAM School of Technology, GITAM, Visakhapatnam, 510419 Emil:* pkaru@gitam.edu.
[4]*Assistant Professor, Department of Computer Science & Engineering, School of Computing Science and Engineering, VIT Bhopal University Gram: Kothari Kalan, Sehore Emil*: *suneetjoshi_2000@yahoo.com.*
[5]*Research Coordinator, Department of Computer Science and Engineering, G H Raisoni University, Nagpur, Email soniabajaj600@gmail.com.*
[6]*Associate professor, Department of Computer Science and Engineering, Saveetha school of Engineering, Saveetha Institute of Medical and Technical sciences (SIMATS) Email : saralav.sse@saveetha.com*

information about these threats. This proactive approach is supported by strong management of keys and user identities, building a robust security system. Regularly updating and fixing any vulnerabilities, along with strictly following rules and standards, is vital for strengthening the overall security of the cloud environment [7].

Having a comprehensive and flexible technology framework is essential to stay ahead of evolving cyber threats. This not only ensures that data remains confidential, intact, and accessible but also establishes a secure and sturdy foundation for cloud infrastructure. As the digital landscape changes, it's extremely important for organizations to adopt advanced security measures to manage the intricacies of cloud security successfully. Old security methods are no match for today's cyber threats. There is a need for smarter solutions to keep your digital world safe and sound. The objectives of the proposed work are:

I. Develop and implement a confidential cloud computing architecture focusing on prioritizing robust security measures.
II. Identify and analyze a spectrum of threats to cloud computing, encompassing data breaches, unauthorized access, insider threats, and emerging cyber threats.
III. Implement strong encryption techniques to safeguard data during both transit and rest periods.
IV. Explore and deploy advanced encryption algorithms and technologies to elevate the overall level of data protection.
V. Develop and integrate stringent access control mechanisms, effectively limiting unauthorized access to sensitive data.

2.Literature Review

2.1 Background and motivation

During the initial stages of cloud security, the protection of data heavily leaned on what are now deemed traditional methodologies. The deployment of firewalls played a pivotal role, serving as the first line of defense by erecting barriers to monitor and control network traffic [8]. Basic security like firewalls and encryption are used to protect data in the cloud. But as cyber threats got more advanced, these methods showed limitations. So, need for smarter and more advanced security strategies is necessary to keep our data safe in today's digital world.

In today's cybersecurity paradigm, the need for sophisticated security measures has become paramount. Recognizing the inadequacies of older methods, organizations are embracing advanced security protocols to counteract the ever-growing spectrum of cyber threats. Modern security methods, such as multi-factor authentication, anomaly detection, and strong encryption, aim to create a flexible defense system. This system adapts to today's complex threats, making cloud security an ongoing process that requires a commitment to using cutting-edge measures against emerging risks.

2.1.1 Firewall

Firewalls as sentinels guarding private networks from external threats. They formed a barrier, scrutinizing incoming and outgoing network traffic according to pre-defined security rules. By filtering data packets, these firewalls prevented unauthorized access and potential cyber-attacks [9]. Traditional firewalls can be perceived as somewhat outdated in cloud data security due to their limited visibility into modern distributed applications, challenges in adapting to dynamic cloud environments, and potential struggles with effectively handling encrypted traffic. As cyber threats evolve, the focus has shifted towards more comprehensive security strategies, including cloud-native security services, micro-segmentation, and continuous monitoring [10], to address the complexities of securing data in the cloud.

2.1.2 Basic Encryption

In the nascent stages of cloud security, reliance on basic encryption methods was commonplace to safeguard data in transit and at rest. These rudimentary encryption techniques played a crucial role in providing a level of protection [11]. However, with the evolution of cyber threats, the limitations of basic encryption became evident. The contemporary digital landscape demands more sophisticated encryption techniques to effectively thwart advanced cyber threats and ensure the comprehensive security of sensitive information stored and transmitted within cloud environments.

2.1.3 Virtual Private Networks

Many use Virtual Private Networks (VPNs) to create secure and encrypted internet connections for accessing cloud resources safely. VPNs worked like private tunnels over the internet, adding a layer of security to prevent unauthorized access [12]. While effective, VPNs have their drawbacks against today's advanced cyber threats, like sophisticated malware and targeted attacks [13]. These threats can take advantage of possible weaknesses in VPNs, emphasizing the importance of adopting more advanced and thorough security measures to ensure strong protection for users accessing cloud resources.

3. Proposed Work

Prioritizing data security, the process initiates with the cloud provider initializing secure enclaves through technologies like Intel SGX or AMD SEV. Users identify sensitive workloads, initiating enclave creation, and encrypt data before entry. Secure deployment to the cloud follows, with user authentication, enabling access to the enclave. Computation occurs within this secure environment, ensuring the confidentiality of encrypted data. Results are generated, encrypted within the enclave, and securely transmitted. At the user end, decryption with appropriate keys occurs in a trusted environment, allowing interaction

with decrypted results. This meticulous process ensures end-to-end data security throughout computations and transmissions.

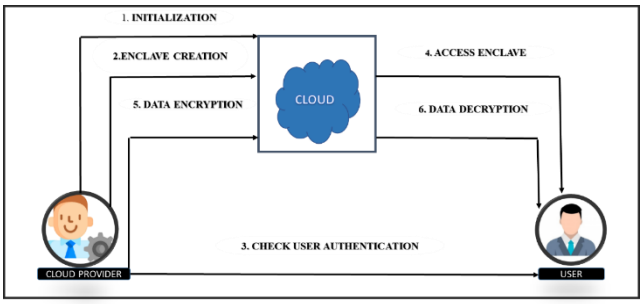

Fig 1. Workflow of Confidential Computation

3.1 Initialization

Initializing enclave creation is a crucial step in implementing confidential computing. This process involves selecting enclaves tailored to the application's security requirements. Identify sensitive operations, data, and processes that require protection. Evaluate existing enclave options or, if necessary, develop custom enclaves that align with your security needs. Enclaves provide isolated and secure execution environments to protect sensitive computations. Depending on your cloud provider, explore hardware-based solutions for confidential computing, such as Intel SGX or AMD SEV. Hardware-based enclaves offer additional security measures by isolating code and data at the processor level.

3.2 Enclave creation

After initialization, the enclave undergoes a crucial configuration phase, tailoring specific parameters to meet the application's security and performance requirements. This includes setting the enclave's memory size, defining processing power, and specifying necessary hardware resources. These configurations establish the foundation for a secure and optimized execution environment. Memory protection mechanisms play a pivotal role in enclave functionality. Enclaves utilize these mechanisms to isolate their code and data from the broader system. Memory pages assigned to the enclave are safeguarded against unauthorized access, creating a secure enclave execution environment that prevents data breaches. Secure key generation and management are fundamental aspects of enclave creation. Enclaves often rely on a Trusted Execution Environment (TEE) to furnish a secure and isolated execution space. Activation of the TEE is a critical step, guaranteeing that the enclave operates within a protected environment. This prevents unauthorized access and tampering, fortifying the enclave against potential security threats. Let S denote sensitive workloads/data, and E represent the process of enclave initialization. The selection of enclaves tailored to security needs can be expressed as:

$$E=f(S) \quad (1)$$

Here, f represents the function or process of choosing or developing enclaves based on the sensitivity of the data/workloads. Consider M as the parameter for memory size, P for processing power, and MPM for memory protection mechanisms. The configuration of enclaves (E_{config}) is a function that configures the parameters for secure enclave creation.

$$E_g= M \times P \times MPM \quad (2)$$

3.3 User authentication

Following enclave creation, the cloud provider enforces stringent authorization measures to prevent unauthorized access. Access control policies are implemented, specifying which users or entities have permission to interact with the enclave. Robust identity verification, often employing multi-factor authentication, ensures the legitimacy of entities seeking access. Role-Based Access Control (RBAC) assigns specific roles and permissions, limiting access to authorized individuals. Through these measures, the cloud provider fortifies the enclave, assuring that only authenticated and authorized entities access and interact with its secure environment, upholding the confidentiality and integrity of sensitive enclave operations and data. Authentication and access control involve verifying entities (V), implementing access policies (AP), and role-based control ($RBAC$)

$$Auth=h(V, AP, RBAC) \quad (3)$$

Here, h represents the authentication and access control process based on entity verification, access policies, and role-based control mechanisms.

3.4 Access enclave

For an authenticated user, access to the enclave is carefully orchestrated by the cloud provider. From authentication and authorization to secure communication and key exchange, each step ensures that only authorized and authenticated users can securely access and interact with the enclave within the cloud infrastructure. The secure communication and access process (A) by authenticated users (AU) can be represented as:

$$A=k\,(AU) \quad (4)$$

Where k represents the process of allowing access A based on authenticated user credentials.

3.5 Data encryption

In the data encryption step of a confidential enclave, sensitive information undergoes robust protection through cryptographic techniques. The process encompasses encrypting data at rest, in transit, and during enclave processing, converting plaintext into unreadable ciphertext without the designated decryption key. The cryptographic

keys responsible for encryption are exclusively owned and securely managed within the enclave, with access restricted to authorized entities, including the enclave itself or designated users possessing the requisite permissions. Strict controls govern decryption key access, ensuring that only authenticated entities, such as the enclave or authorized users, can decipher and access the original data. Encryption of data (*D*) within the enclave can be expressed as cryptographic operation:

$$D = E_t(D,K) \quad (5)$$

Here, *E* represents the encryption function using encryption key *K*.

3.6 Data decryption

Data decryption within enclaves is a pivotal step in ensuring confidential processing. This involves implementing robust mechanisms for secure decryption and safeguarding sensitive information during data operations. Regular security audits play a crucial role in identifying vulnerabilities and monitoring enclave activities, ensuring a proactive approach to security. By conducting periodic assessments, potential security incidents can be promptly addressed, fortifying the enclave's resilience against emerging threats.

$$D = D_t(D_e,K) \quad (6)$$

Here, D_c represents the decryption function using decryption key *K*

3.7 Algorithmic Framework

Algorithm: Enclave Creation in Confidential Computing

```
#Initialize Enclave
enclave_key = secrets.token_bytes(32)
#Generate a hash-based key
return enclave_key, integrity_key
create_enclave(data, partition_size, threshold):
enclave_key, integrity_key = generate_keys()
#Divide data into data partitions
  for i in range(0, len(data), partition_size)]
# Initialize variables
  repetition_count = 0
  aggregated_data = []
#Iterate over each partition
 for partition in partitions:
  repetition_count += 1
  partition_data = []
#Process each attribute in the partition
  for attribute in partition:
  # Additional steps for merging
  partition_data = merge(partition_data, attribute)
#Merge all attributes to generate partition result
 partition_result = merge_all_aos(partition_data)
#Check for repetition
 if repetition_count == threshold:
#Apply necessary privacy-enhancing operations
  partition_result =
privacy_enhancing_operations(partition_result)
#Release partition_result - Perform secure release
  release_data(partition_result)
# Reset repetition counter
  repetition_count = 0
Additional steps if threshold == -1
 if threshold == -1:
# Apply necessary privacy-enhancing operations
  partition_result =
privacy_enhancing_operations(partition_results
# Release partition_result - Perform secure release
  release_data(partition_result)
 # Return the final enclave data and keys
  return partition_result, enclave
```

The above algorithm outlines the process of initializing and creating a secure enclave within a cloud environment that supports confidential computing. In the initialization phase, a cryptographic key (`enclave_key`) is generated using a secure method (`secrets.token_bytes(32)`), ensuring a 32-byte random key for securing the enclave. Additionally, a hash-based integrity key (`integrity_key`) is generated. The `create_enclave` function takes parameters such as `data`, `partition_size`, and `threshold`. It begins by generating enclave and integrity keys using the `generate_keys` function. The data is then divided into partitions of size `partition_size`, and key variables, including `repetition_count` and `aggregated_data`, are initialized. The algorithm iterates over each partition, incrementing the repetition count and initializing partition-specific data. It processes each attribute within the partition, incorporating additional steps for merging the data. The merged attributes form a partition result. The algorithm checks for repetition based on the `threshold`. If the repetition count equals the threshold, necessary privacy-enhancing operations are applied to the partition result. The result is then securely released using `release_data`. The repetition counter is reset to zero. If the threshold is set to -1, indicating no specific limit, additional privacy-enhancing operations are applied, and the partition result is securely released. The final output includes the enclave's partitioned result and the enclave key. This algorithm provides a clear and concise methodology for creating secure enclaves, emphasizing data privacy and integrity within a confidential computing-supported cloud environment.

4.Results

Confidential computing establishes secure enclaves within cloud environments, ensuring that sensitive data is processed

in isolated, encrypted spaces. This significantly reduces the risk of unauthorized access, providing a heightened level of data confidentiality compared to traditional security measures. With confidential computing, even cloud service providers cannot access the data within secure enclaves. This safeguards against potential insider threats, ensuring that only authorized users can access sensitive information, bolstering data security, and mitigating risks associated with internal actors. Confidential computing allows for the secure processing of encrypted data. This means that even when data is in use, it remains encrypted within the enclave, providing an additional layer of protection. This is particularly crucial in scenarios where sensitive information needs to be analyzed or processed without compromising its encryption, reducing the attack surface for potential threats.

4.1 System Specification

The foundational component, CloudSim 3.0.3, forms the core for simulating and assessing performance. Operating on Windows 7 ensures seamless compatibility. Integration with Apache Common-math becomes pivotal for mathematical operations within CloudSim. The desktop's robust configuration, boasting 8 GB RAM, 1 TB storage, and a 3.40 GHz Intel Core i5 processor, aligns with optimal performance standards. A Java Runtime Environment compatible with CloudSim's Java-based structure is indispensable. Configuring VM specifications, including 10 data centers, 2000 nodes, and 1 GB/s network bandwidth, establishes a heterogeneous cloud environment. Defined network bandwidth (1 GB/s) and storage capacity (4 GB) adhere to simulation prerequisites.

4.2 Dataset Description

The dataset, named "CryptoBenchmark_KaggleSample," sourced from Kaggle is of moderate size, comprising five data points. Each data point corresponds to a unique user count ranging from 10 to 50. The dataset provides a comparative analysis of key generation and retrieval times in general computation versus confidential computing across varying user counts. It demonstrates the consistent superiority of confidential computing in cryptographic tasks. As user counts increase, the dataset showcases the amplified efficiency of confidential computing, delivering reduced key generation times compared to conventional methods. Notably, it emphasizes the robust performance of confidential computing in key retrieval operations, affirming its supremacy in securely accessing cryptographic keys. These insights underscore the potential benefits of adopting confidential computing, particularly in scenarios with larger user counts, where its performance advantages become increasingly evident. Table 1 provides the comparison of key generation and retrieval time for varying user count.

Table 1. Comparison of Crypto-Performance: Key Generation and Retrieval Time

User Count	General Computation Key Generation (ms)	Confidential Computing Key Generation (ms)	General Computation Key Retrieval (ms)	Confidential Computing Key Retrieval (ms)
10	14	8	20	9
20	20	10	25	12
30	17	9	22	10
40	22	11	27	13
50	19	10	24	11

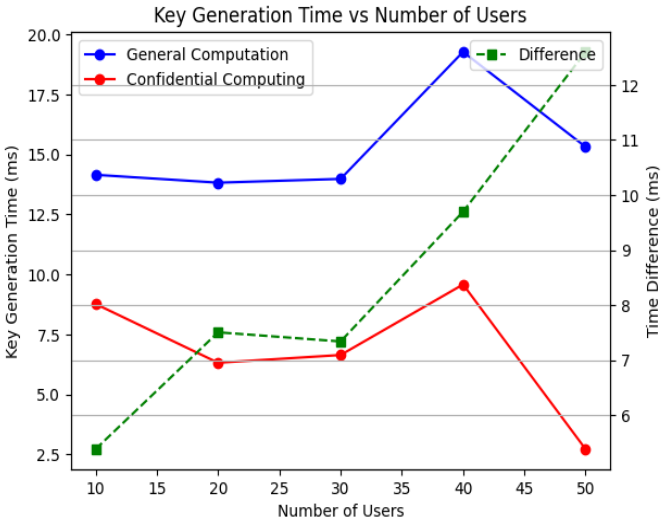

Fig. 2. Key generation time vs. number of users

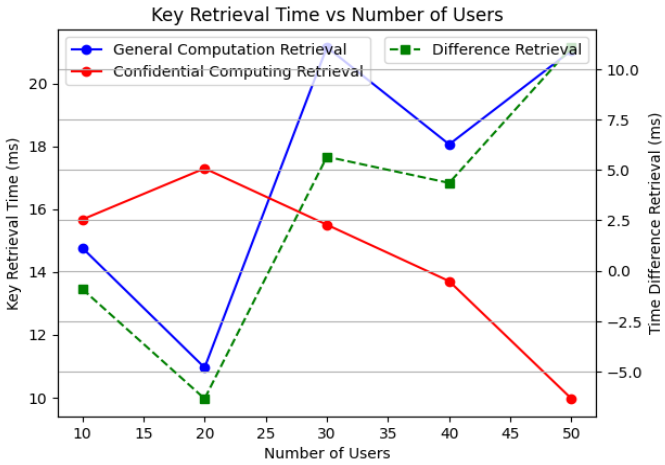

Fig. 3. Key retrieval time vs. number of users

In Fig.2 and Fig.3, The graph includes two lines representing the key generation times for general computation (blue line) and confidential computing (red line). Additionally, a second y-axis is added to show the

difference in key generation times (green dashed line). Here's how you can interpret the graph.

i. If the red line (confidential computing) consistently stays below the blue line (general computation), it suggests that confidential computing generally outperforms general computation in terms of key generation times.

ii. If the green dashed line consistently stays above zero, it indicates that confidential computing consistently has a positive performance difference, reinforcing the idea that it's better.

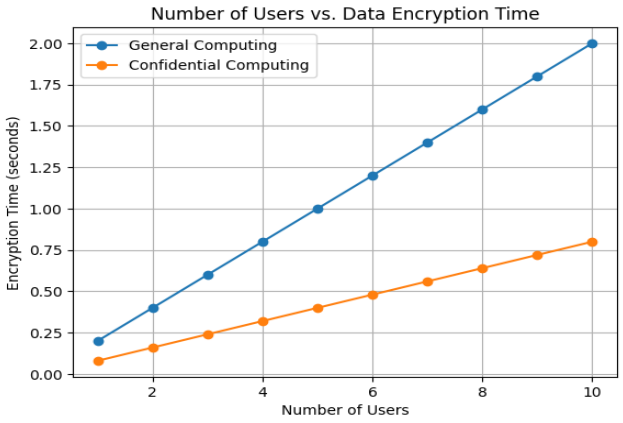

Fig. 4. Number of users vs. data encryption time

From Fig.4, the x-axis (increase the number of users), you can observe the corresponding values on the y-axis (encryption time). The blue line (general computing) shows a linear increase in encryption time, following the relationship defined in the placeholder function. The yellow line (confidential computing) shows a consistently lower encryption time compared to general computing. This implies that, based on the placeholder scenarios, confidential computing performs encryption more efficiently than general computing.

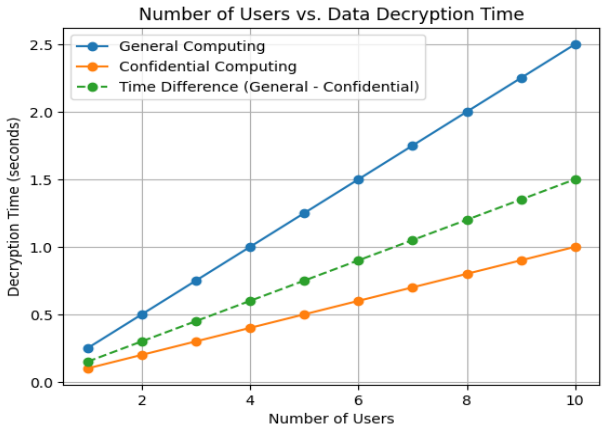

Fig. 5. Number of users vs. data decryption time

In Fig.5, As the number of users increases, both general computing and confidential computing experience an increase in decryption time, as indicated by the blue and orange lines, respectively. The dashed green line (Time Difference) helps emphasize the advantage of confidential computing. Positive values of the time difference indicate that general computing takes more time for decryption compared to confidential computing.

5.Conclusion and Future Scope

The comparative analysis of key generation times between general computation and confidential computing, as depicted in the graph, unequivocally positions confidential computing as the superior choice for ensuring robust data security in a cloud platform. The consistent advantage of confidential computing, represented by the red line consistently residing below the blue line, signifies its efficiency in cryptographic key generation. This not only showcases the reliability of confidential computing but also underscores its paramount role in enhancing data security within the dynamic landscape of cloud platforms. The positive trajectory of the green dashed line further reinforces the notion that confidential computing consistently outshines general computation. As organizations navigate the intricate realm of cloud security, the evidence presented strongly advocates for adopting confidential computing as the optimal solution for fortifying data security in cloud environments. Ongoing advancements in confidential computing technologies are anticipated, offering improved security features and performance optimizations. The future scope lies in seamlessly integrating confidential computing with emerging cloud architectures, ensuring robust data security in evolving cloud environments.

References

[1] Cheng, L., Liu, F., & Yao, D. (2017). Enterprise data breach: causes, challenges, prevention, and future directions. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 7(5), e1211.

[2] Sharma, N., & Shamkuwar, M. (2019). Big data analysis in cloud and machine learning. Big

[3] Bandari, V. (2023). Enterprise Data Security Measures: A Comparative Review of Effectiveness and Risks Across Different Industries and Organization Types. International Journal of Business Intelligence and Big Data Analytics, 6(1), 1-11.

[4] Mather, T., Kumaraswamy, S., & Latif, S. (2009). Cloud security and privacy: an enterprise perspective on risks and compliance. " O'Reilly Media, Inc.".

[5] Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2013). Network anomaly detection: methods, systems and tools. Ieee communications surveys & tutorials, 16(1), 303-336.

[6] Aziz, R., Banerjee, S., Bouzefrane, S., & Le Vinh, T. (2023). Exploring Homomorphic Encryption and Differential Privacy Techniques towards Secure Federated Learning Paradigm. Future internet, 15(9), 310.

[7] Kumar, R., & Goyal, R. (2019). On cloud security requirements, threats, vulnerabilities and countermeasures: A survey. Computer Science Review, 33, 1-48.
[8] Arogundade, O. R. (2023). Network Security Concepts, Dangers, and Defense Best Practical. Computer Engineering and Intelligent Systems, 14(2).
[9] Uçtu, G., Alkan, M., Doğru, İ. A., & Dörterler, M. (2021). A suggested testbed to evaluate multicast network and threat prevention performance of Next Generation Firewalls. Future Generation Computer Systems, 124, 56-67.
[10] Dhayanidhi, G. (2022). Research on IoT Threats & Implementation of AI/ML to Address Emeging Cybersecurity Issues in IoT with Cloud Computing.
[11] Zulifqar, I., Anayat, S., & Kharal, I. (2021). A Review of Data Security Challenges and their Solutions in Cloud Computing. International Journal of Information Engineering & Electronic Business, 13(3).
[12] Nyakomitta, P. S., & Abeka, S. O. (2020). Security investigation on remote access methods of virtual private network. Global journal of computer science and technology, 20.
[13] Sapalo Sicato, J. C., Sharma, P. K., Loia, V., & Park, J. H. (2019). VPNFilter malware analysis on cyber threat in smart home network. Applied Sciences, 9(13), 2763.

Find features set Fs = $(\sum_{j=1}^{size(Ts)} Features \in Ts(i)) \cup Fs$

For each record r of Cd

Add features of Fs with r

R = $((\sum Features \in r) \cup \sum Features(Fs)$

For each feature f belongs to Fs in R

R(Fs(f))= Random(Fs(f).range_min, Fs(f).range>max)

End

End

End

End

Stop

The feature level normalizer algorithm finds the features of other data sets and computes Range_min and Range_max values. According to the range values, the method generates number of records and appends the features of other data set and initializes them with different random values between the ranges computed. The preprocessed set is used to perform intrusion detection.

The above algorithm computes feature impact frequency for variety of features on the different data set. As per frequency values, a subset of features are identified to perform intrusion detection.

3.3 DNN Training

The method trains the deep neural network with number of intermediate layers. The number of intermediate layer is decided according to the number of classes and number of trust values measured. Accordingly, the model trained with six layers with four intermediate layers. The first intermediate layers involve in computing Feature Level Trust (FLT) and Transmission Level Trust Weight (TLTW) for the binary class with 1, where the second two intermediate layers are designed to compute Feature Level Trust (FLT) and Transmission Level Trust Weight (TLTW) values for the binary class 0. The output layer returns two set of Feature Level Trust (FLT) and Transmission Level Trust Weight (TLTW) values which has been used to compute MCTW value to perform intrusion detection.

3.4 DNN Testing

The test sample given has been taken for DNN testing. With the tuple given, the method extracts the features and generates the feature vector. Generated feature vector has been passed to the network trained. The first intermediate layers involve in computing Feature Level Trust (FLT) and Transmission Level Trust Weight (TLTW) for the binary class with 1, where the second two intermediate layers are designed to compute Feature Level Trust (FLT) and Transmission Level Trust Weight (TLTW) values for the binary class 0. Obtained result on the output layer has been used to compute MCTW value. Estimated value has been used to perform intrusion detection.

Algorithm:

Given: DNN, Test Sample Ts

Obtain: Class C

Start

Read DNN and Ts.

Feature vector fv = $\sum Features \in Ts$

Pass Fv to DNN.

At first intermediate layer

Each neuron computes Feature Level Trust Flt for genuine class.

$$\text{FLT (gc)} = \frac{\sum\left(\frac{\sum_{j=1}^{size(Fv)} Count(Dst(Fv(j),Cs(i)(j))<Th)}{size(Fv)}\right)}{\sum_{sizei=1}^{size(Cs)}}$$

At second intermediate layer.

Each neuron computes Tltw value for genuine class.Compute Transmission Level Trust Weight Tltw.

$$\text{Tltw (gc)} = \frac{\sum_{size(i=1)}^{size(Cs)} Count(Cs(i).state==Genuine)}{Size(Cs)}$$

Third intermediate layer computes FLT for malicious class.

$$\text{FLT (mc)} = \frac{\sum\left(\frac{\sum_{j=1}^{size(Fv)} Count(Dst(Fv(j),Cs(i)(j))<Th)}{size(Fv)}\right)}{\sum_{sizei=1}^{size(Cs)}}$$

Fourth intermediate layer computes Tltw value for malicious class.

$$\text{Tltw (Mc)} = \frac{\sum_{size(i=1)}^{size(Cs)} Count(Cs(i).state==Genuine)}{Size(Cs)}$$

Output layer returns Flt(gc)Tltw(gc),Flt(mc) and Tltw(mc).

Compute MLTW value for genuine class

MLTW(gc) = FLT(gc)×TLTW(gc).

Compute MLTW value for malicious class

MLTW(mc) = FLT(gc)×TLTW(gc).

Class C= choose the class value with maximum MLTW value

Stop

The DNN testing algorithm computes MLTW value for various classes and based on that the method identifies the class of data.

4. Results and Discussion

The proposed model is implemented using matlab and performance is measured using different data sets. The performance evaluation is carried out by using NSL-KDD, UNSW-NB15 and AWID data sets.

Classification Accuracy:

The performance of the method is measured for its classification accuracy. It has bee measured as follows:

$$\text{Classification Accuracy} = \frac{\text{TP+TN}}{\text{TP +TN+FP+FN}}$$

Precision:

The precision represent the positive rate produced by the method in classification. It has been measured as follows:

$$\text{PR} = \frac{TP}{TP+FP}$$

Recall:

Recall is the measure which represents the true positive rate produced by the method. It has been measured as follows:

$$\text{TPR} = \frac{\text{TP}}{\text{TP +FN}}$$

False Positive Rate:

FPR is the measure which represents the ratio of false classification produced by the method. It has been measured as follows:

$$\text{FPR} = \frac{\text{FP}}{\text{FP +TN}}$$

According to the above factors, the methods are measured for their performance and presented in this section.

Table 1: Analysis on various metrics

Methods	TPR (%)	FPR (%)	Precision (%)	Accuracy (%)
LGTBIDS	89	5	81.65	93.80
TCE-IDS	75	8.75	68.18	88
FLAG	68	11.75	59.13	84.20
CNN-GRU	80	6.25	76.19	91
RNN-LSTM	99	2.3	98.9	99
RFIO-DNN	99.2	1.4	99.4	99.6

The performance of the method is evaluated on different metrics and displayed in Table 1. The proposed RFIO-DNN method introduces higher performance in all the factors.

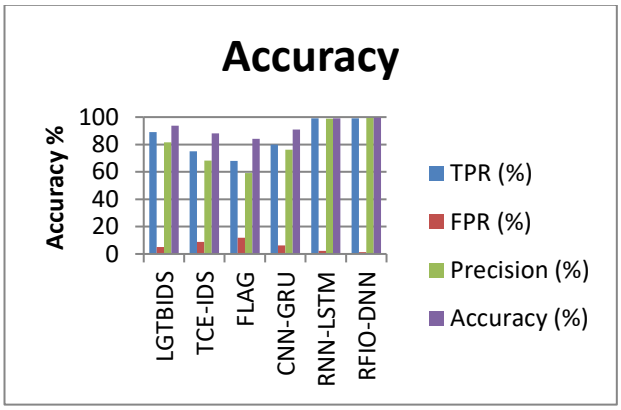

Fig 2. Performance Analysis

Analysis of various metrics is performed and compared in Figure 2. The proposed, RFIO-DNN algorithm has produced higher performance in all the factors. The method used NSL-KDD and UNSW-NB 15 data sets. Both were merged and normalized to frame the data set and features are extracted to train the model. Accordingly, the method are measured for their performance and presented in the above Figure 2. In all the case, the RFIO-DNN algorithm has produced higher performance than others.

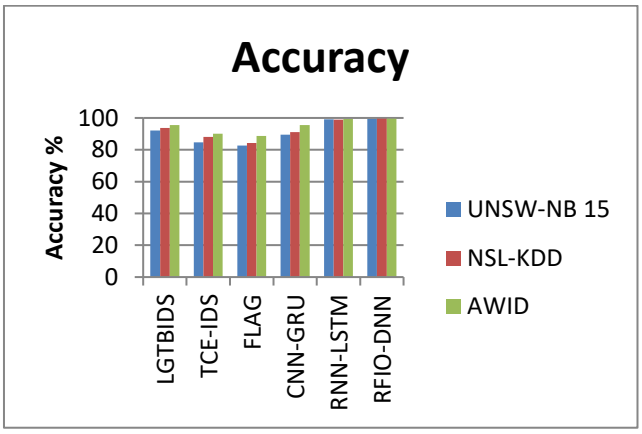

Fig 3: Analysis on Accuracy

The performance in classification accuracy is measured for different data sets and presented in Figure 3. The proposed RFIO-DNN algorithm has produced higher accuracy in classification than other methods.

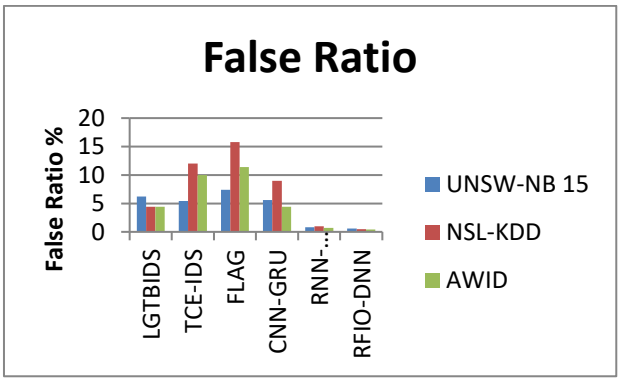

Fig 4: Analysis on false ratio

The ratio of false classification produced by different methods are measured and presented in Figure 4. The proposed RFIO-DNN algorithm has produced less false classification than others.

5.Summary

This paper presented a novel realtime feature impact optimization based DNN model (RFIO-DNN) towards intrusion detection in 5G networks. The method uses various data sets and merge them towards normalization using Feature Level Fuzzy Normalizer. Further, Feature Impact Optimization (RFIO) algorithm is applied to select specific features from the data set. Then, the features and values of the records is extracted and converted into feature vector. Extracted feature vector set has been used to train the deep neural network. At the test phase, the neurons of the network compute Feature Level Trust (FLT) and Transmission Level Trust Weight (TLTW). Using these values, the output layer neuron computes Multi Constraint Trust Weight (MCTW) according to the IoT devices present in the transmission route. Using the value of MCTW, the method classifies the incoming data as well as node to perform intrusion detection.

References

[1] A. Pathak, I. Al-Anbagi and H. J. Hamilton, "An Adaptive QoS and Trust-Based Lightweight Secure Routing Algorithm for WSNs," in IEEE Internet of Things Journal, vol. 9, no. 23, pp. 23826-23840, 1 Dec.1, 2022, doi: 10.1109/JIOT.2022.3189832.

[2] S. Verma, S. Zeadally, S. Kaur and A. K. Sharma, "Intelligent and Secure Clustering in Wireless Sensor Network (WSN)-Based Intelligent Transportation Systems," in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 8, pp. 13473-13481, Aug. 2022, doi: 10.1109/TITS.2021.3124730.

[3] A. Ahmed, S. Abdullah, M. Bukhsh, I. Ahmad and Z. Mushtaq, "An Energy-Efficient Data Aggregation Mechanism for IoT Secured by Blockchain," in IEEE Access, vol. 10, pp. 11404-11419, 2022, doi: 10.1109/ACCESS.2022.3146295.

[4] M. Bin-Yahya, O. Alhussein and X. Shen, "Securing Software-Defined WSNs Communication via Trust Management," in IEEE Internet of Things Journal, vol. 9, no. 22, pp. 22230-22245, 15 Nov.15, 2022, doi: 10.1109/JIOT.2021.3102578.

[5] M. Nouman, U. Qasim, H. Nasir, A. Almasoud, M. Imran and N. Javaid, "Malicious Node

Detection Using Machine Learning and Distributed Data Storage Using Blockchain in WSNs," in IEEE Access, vol. 11, pp. 6106-6121, 2023, doi: 10.1109/ACCESS.2023.3236983.

[6] Z. Yu, Y. Liu, G. Xie, R. Li, S. Liu and L. T. Yang, "TCE-IDS: Time Interval Conditional Entropy-Based Intrusion Detection System for Automotive Controller Area Networks," in IEEE Transactions on Industrial Informatics, vol. 19, no. 2, pp. 1185-1195, Feb. 2023, doi: 10.1109/TII.2022.3202539.

[7] K. Sood, M. R. Nosouhi, D. D. N. Nguyen, F. Jiang, M. Chowdhury and R. Doss, "Intrusion Detection Scheme With Dimensionality Reduction in Next Generation Networks," in IEEE Transactions on Information Forensics and Security, vol. 18, pp. 965-979, 2023, doi: 10.1109/TIFS.2022.3233777.

[8] M. Shafi, R. K. Jha and S. Jain, "LGTBIDS: Layer-Wise Graph Theory-Based Intrusion Detection System in Beyond 5G," in IEEE Transactions on Network and Service Management, vol. 20, no. 1, pp. 658-671, March 2023, doi: 10.1109/TNSM.2022.3197921.

[9] Z. Zhao, Q. Du and H. Song, "Traffic Load Learning Towards Early Detection of Intrusion in Industrial mMTC Networks," in IEEE Transactions on Industrial Informatics, vol. 19, no. 7, pp. 8441-8451, July 2023, doi: 10.1109/TII.2022.3218722.

[10] H. Whitworth, S. Al-Rubaye, A. Tsourdos and J. Jiggins, "5G Aviation Networks Using Novel AI Approach for DDoS
Detection," in IEEE Access, vol. 11, pp. 77518-77542, 2023, doi: 10.1109/ACCESS.2023.3296311.

[11] D. D. N. Nguyen, K. Sood, Y. Xiang, L. Gao, L. Chi and S. Yu, "Toward IoT Node Authentication Mechanism in Next Generation Networks," in IEEE Internet of Things Journal, vol. 10, no. 15, pp. 13333-13341, 1 Aug.1, 2023, doi: 10.1109/JIOT.2023.3262822.

[12] T. Ye, G. Li, I. Ahmad, C. Zhang, X. Lin and J. Li, "FLAG: Few-Shot Latent Dirichlet Generative Learning for Semantic-Aware Traffic Detection," in IEEE Transactions on Network and Service Management, vol. 19, no. 1, pp. 73-88, March 2022, doi: 10.1109/TNSM.2021.3131266.

[13] Y. He, M. Kong, C. Du, D. Yao and M. Yu, "Communication Security Analysis of Intelligent Transportation System Using 5G Internet of Things From the Perspective of Big Data," in IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 2, pp. 2199-2207, Feb. 2023, doi: 10.1109/TITS.2022.3141788.

[14] M. Lefoane, I. Ghafir, S. Kabir and I. -U. Awan, "Unsupervised Learning for Feature Selection: A Proposed Solution for Botnet Detection in 5G Networks," in IEEE Transactions on Industrial Informatics, vol. 19, no. 1, pp. 921-929, Jan. 2023, doi: 10.1109/TII.2022.3192044.