INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

e-Healthsec: A Cloud-Based Privacy-Preserving Electronic Health History Framework using NLP with Multi-Layer Encryption

N Subhalakshmi[1], M V Srinath[2]*

1 Research Scholar, S.T. E. T Women's College (Autonomous), (Affiliated to Bharathidasan University, Tiruchirappalli), Sundarakkottai, Mannargudi - 614016, Thiruvarur Dt., Tamil Nadu, India
2 Research Supervisor, S.T. E. T Women's College (Autonomous), (Affiliated to Bharathidasan University, Tiruchirappalli), Sundarakkottai, Mannargudi - 614016, Thiruvarur Dt., Tamil Nadu, India

OPEN ACCESS

Received: 08-01-2025
Accepted: 10-02-2025
Published: 21-02-2025

Citation: Subhalakshmi N, Srinath MV (2025) e-Healthsec: A Cloud-Based Privacy-Preserving Electronic Health History Framework using NLP with Multi-Layer Encryption. Indian Journal of Science and Technology 18(6): 415-429. https://doi.org/10.17485/IJST/v18i6.51

* **Corresponding author**.

subha.stet@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Background: An effective protection of the large volume of health data and Electronic Health History (EHHs) is a significant challenge in cloud. Physicians need patient's medical records, including multimedia big data analytics to diagnose patient. Online EHH allows patients to efficiently manage their health records. This can be efficiently achieved by using cloud computing to store EHH. **Objectives:** The research aims to preserve patients' privacy by efficiently protecting their sensitive EHH in cloud storage by developing a multi-layered encryption framework. **Methods:** EHHs are encrypted using Fully Homomorphic Encryption (FHE) and Fernet encryption method before being stored in the cloud environment. The proposed system in this work is divided into three modules such as EHH storage after NLP; Key management and authentication; and data storage on the Amazon Web Services (AWS) cloud. **Findings:** EHH Encryption framework: e-Healthsec using Fernet and FHE which provides multi-layer security for sensitive EHHs.The comparative analysis of performance metrics such as bandwidth, memory usage and level of protection through different encryption techniques determines the Quality of Service (QoS). **Novelty and applications:** A novel EHH Framework - e-Healthsec that leverages the influence of NLP and text mining with EHH Encryption algorithm to protect sensitive health records.Thusenablesthe healthcare practitioners to control access to patient records while processing and analyzing critical data.

Keywords: Electronic Health History; Cloud Computing; Fernet; Natural Language Processing (NLP); Fully Homomorphic Encryption (FHE)

1 Introduction

India's healthcare system confronts major hurdles, owing to inadequate usage of technology and fragmented patient data. Patients frequently see many doctors from

diverse disciplines, necessitating that each doctor have access to a complete medical history, which is challenging to obtain and manage. This lack of integrated medical information can result in missed diagnosis and potential risks to patient safety. The existing healthcare system frequently results in inefficient and potentially harmful practices, such as incorrect medicine prescriptions. Cloud computing provides a solution by providing safe data storage, management, and analysis, hence increasing the overall efficiency of healthcare services [1].

Big data in healthcare refers to large amounts of complex medical data that cannot be managed using typical software or equipment. Transferring patient information, such as test results and medical histories, to healthcare professionals over the internet is insecure. Cloud technologies address this by securely storing all patient records, allowing for simple access across many hospitals without the need to transport physical paperwork. Big data analytics can revolutionize healthcare by enhancing decision-making, patient outcomes, and operational efficiency [2].

As data volumes grow, large data storage becomes increasingly important, with cloud storage making data available from anywhere. Cloud computing, which provides benefits such as cost-effectiveness, scalability, and storage, is driving enterprises to migrate from local to remote cloud servers for data management [3]. Healthcare cloud systems facilitate communication among service providers, clinics, and patients, but they confront substantial problems such as security, privacy, and regulatory concerns. To address these problems, novel ways are required to protect the security and privacy of electronic health records [4,5].

Storing medical data on the cloud computing platform permits users to store their data conveniently. Personal data is first encoded by the user and directed to cloud storage. The encoded data is reorganized by the cloud platform after being stripped of important new data. Encoded cloud data is typically managed by a third party [6,7]. Subsequently, the integrity, privacy, and safety of health record data are less secure than private storing systems. Consequently, it is significant to ensure that unauthorized third parties can access or modify encoded cloud data. Cloud platforms may use cryptographic methods with private keys generated by third parties wherever third parties are responsible for cloud storage. Another problem is that unauthorized users can search data, resulting in data leakage to strangers.

This research delivers a protected data access scheme that provides greater security and integrity when storing and sharing sensitive health care data in the cloud. This system consists of three different cloud platform techniques to enhance the security, privacy, and integrity of users' personal data.

The main objectives of this research are:

This research aims to provide an enhanced security for cloud computing platforms where encrypted data is changed without the original data awareness and authorized users recover and decode the data. This is a privacy mechanism that maintains the privacy of sensitive data over the user's verification process.

This work provides a brief overview of protecting confidential healthcare big data on cloud. The usage of big data is attracting attention to the architecture and technology will continue to help us cope with the rapid growth of data in the healthcare applications. This research focus on the novel design and development of intelligent and safe healthcare information system using NLP (spell check, capitalisation, tokenisation and POS tagging and notability detection) with a novel EHH encryption. Large amounts of data from the medical history can be handled using advanced security techniques. The novelty is in training the best data safety layers and storage practices to maintain privacy and security. The data are stored in cloud with EHH encryption algorithm to access the medical history anywhere. A proposed three-layer model seems to be a more efficient big data system for diagnosing diseases and determining patients' health histories as needed. Finally, comparative analysis of QoS parameters such as bandwidth, memory usage and level of protection for proposed EHH with the existing system determine the best fit model for securing the medical history of patients.

Raghavendra Ganiga et al., [8] examined the issues and solutions for secure electronic health records (EHR) in cloud computing systems. The authors suggest a security paradigm tailored to protecting sensitive patient data hosted in the cloud. But there is no comprehensive method that incorporates all required security measures in the context of EHR systems. Pushkar Dubey [9] focused on how AWS can support remote medical monitoring by providing scalable infrastructure, secure data management, and advanced analytics. It might highlight the advantages of leveraging AWS to better patient care and healthcare delivery, as well as address important industry concerns and opportunities. But there is a lack of advanced security mechanism in this work. S. Neela et al., [10] deals about allowing permissions to limit admission to data. Second, it provides a user-centric method to proactively avoid unpredicted user actions on the AWS cloud. This exhibits some degree of resistance at complex levels as it cannot deal with active threats, including new and future threats.

Taranjot Singh [11] examined the Amazon Web Services (AWS)'s impact on the cloud computing sector, taking into account its influence on enterprises, technological innovation, and cloud technology adoption. The vendor lock-in, cost management, security challenges are not clearly stated in this work. Abhijit Maliet al., [12] focused on many security features provided by AWS, how these features assist enterprises in protecting their data and systems in the cloud, and best practices for maintaining robust security in AWS environments.The work describe broad AWS security features and best practices, but it may not go

in-depth into the specific, sophisticated threats that are emerging in cloud systems. P Ramesh Naiduet al.,[13] proposed the multi-tiered framework design effectively prevents cyber attacks and safeguards patients' medical data. The work lacks in data preprocessing which is significant to protect data in cloud. Elias Hossain et al.,[14] provide a thorough evaluation of the role of Natural Language Processing (NLP) in electronic health records (EHRs) for healthcare decision making. It demonstrates how NLP may extract useful information from unstructured clinical data to help with diagnosis, therapy planning, and patient monitoring. The review reveals important obstacles, including data quality, privacy concerns, and integration issues. The work fails to provide a solution to data security.

Gong et al.,[15] work includes a detailed assessment of realistic solutions in fully homomorphic encryption (FHE), as well as an analysis of various acceleration approaches. It investigates the issues of optimizing FHE for practical usage, such as compute efficiency and latency reduction. The issues in using FHE in cloud environment are not properly addressed. L. d. S. Dourado et al.,[16] assesses the performance of big data applications across many cloud providers. It investigates the efficiency, scalability, and resource management capabilities of cloud platforms for large-scale data processing workloads. The analysis reveals which cloud providers perform best for big data applications, allowing enterprises to make more informed decisions regarding their cloud infrastructure. This work lacks to state the encryption mechanism for sensitive healthcare data. Billa Hemanth Venkatesh et al.,[17] present a cloud-based Personal Health Record (PHR) management system integrated with a medical recommender system. To improve data security, researchers could look into novel encryption algorithms, privacy-preserving technologies and advanced access control mechanisms. Thanh Ngoc Nguyen et al.,[18] present an examination of authentication and encryption algorithms for data security in cloud computing. It covers several cryptographic techniques, such as symmetric and asymmetric encryption algorithms, and discusses how they might be used to secure cloud data. A proper encryption technique has notbeen suggested with implementation. Ali S.[19] highlights the significance of deep learning algorithms for detecting schizophrenia using EEG data and proposes a novel paradigm that could have a significant influence on clinical practice and mental health research. Kaur J et al.,[20] address these issues and ensure the security and rapid retrieval of healthcare information, this paper proposes a framework that combines a dual-channel blockchain architecture with two robust cryptographic algorithms, namely Rivest-Shamir-Adleman (RSA) and Advanced Encryption Standard (AES).Wenhua Z et al.,[21] develops a foundation for assuring secure and efficient patient data access control within telehealth systems and established security model as a dependable and efficient solution in the dynamic environment of telemedicine. In conclusion, while numerous studies provide significant insights into the security and administration of healthcare data in cloud systems, several gaps remain, particularly in terms of complete security mechanisms, modern encryption techniques, and tackling growing risks in cloud environments. Additional research and development of secure cloud architectures targeted to healthcare demands is required. The proposed work aims to develop a multi-layer encryption framework that combines Fully Homomorphic Encryption (FHE) and Fernet encryption, addressing the gaps identified in previous studies.

2 Methodology

The proposed architecture diagram of the Electronic Health History (EHH) Encryption framework (e-Healthsec) is shown in Figure 1. In this model, users transmit, and store encrypted data in the cloud. Both provide safe data storage as well as data security during transmission. The information is handled by cloud computing service providers, but it is not readily accessible in plain text.

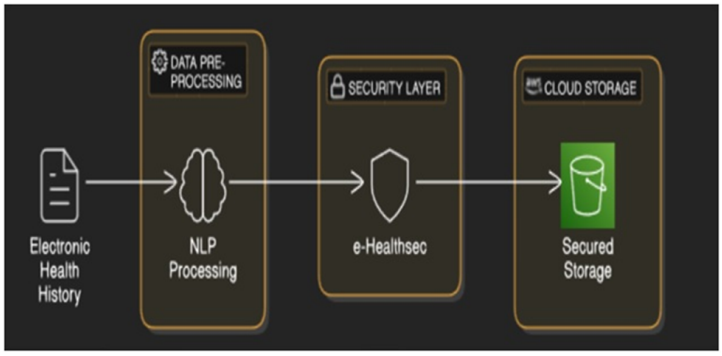

Fig 1. Proposed EHH Encryption Framework : e-Healthsec

2.1 Data Preprocessing

EHH encryption mechanism allows users or reliable third parties to work with ciphertext information directly in place of the original information. The user can get the arithmetic result to decode to get the proper data. For example, in a medical information system, EHH are stored in a cloud server in an encrypted state. When addressing possible safety issues, health departments need to be aware of some areas of specific disease and age distribution. Encrypted EHH data can be provided to expert data processing facilities. The specific information can then be obtained during decryption.

Dataset

The Patient information contains medical, demographic, and over-all data from the record. About 7,000 patients were treated for heart, cancer and tumour diseases. These records comprise particulars about patients namely name, age, gender, family history of illness, laboratory test results, physical examinations by physicians, and screening results. Severity disease changes into acute, chronic, and chronic conditions. Most patients wanted to identify their health status and get prescriptions from an isolated place. Therefore, the data should be stored on the cloud platform. Security and privacy are challenges as patient data is sensitive and non-sensitive. It uses identity encryption together with FHE and Fernet to overcome security and privacy problems. Figure 2 shows the sample dataset consisting of medical history of 23-year-old patients stored as text file. Table 1 shows the sample Dataset consisting of Medical History of Patients. (Source: https://www.kaggle.com)

Input: Medical History (data)

SUBJECTIVE: This 23-year-old white female presents with a complaint of allergies. She used to have allergies when she lived in Seattle but she thinks they are worse here. In the past, she has tried Claritin and Zyrtec. Both worked for a short time but then seemed to lose effectiveness. She has used Allegra also. She used that last summer and she began using it again two weeks ago. It does not appear to be working very well. She has used over-the-counter sprays but no prescription nasal sprays. She does have asthma but does not require daily medication for this and does not think it is flaring up.

MEDICATIONS: Her only medication currently is Ortho Tri-Cyclen and the Allegra.

ALLERGIES: She has no known medicine allergies.

OBJECTIVE:

- **Vitals:** Weight was 130 pounds and blood pressure 124/78.
- **HEENT:** Her throat was mildly erythematous without exudate. Nasal mucosa was erythematous and swollen. Only clear drainage was seen. TMs were clear.
- **Neck:** Supple without adenopathy.
- **Lungs:** Clear.

ASSESSMENT: Allergic rhinitis.

PLAN:

1. She will try Zyrtec instead of Allegra again. Another option will be to use loratadine. She does not think she has prescription coverage so that might be cheaper.
2. Samples of Nasonex two sprays in each nostril given for three weeks. A prescription was written as well.

Fig 2. Sample Dataset Consisting of Medical History of a 23-year-old Patient

Table 1. Sample Dataset Consisting of Medical History of Patients

Patient ID	Name	Age	Gender	Diagnosis	Date of Diagnosis	Treatments	Medications	Allergies	Lab Results	Last Visit Date	Emergency Contact
P001	John Doe	45	Male	Type 2 Diabetes	2022-01-15	Insulin, Lifestyle Changes	Metformin, Insulin	None	Blood Sugar: 150 mg/dL	2023-12-10	Jane Doe (wife)
P002	Mary Smith	60	Female	Hypertension, Arthritis	2021-06-22	Blood Pressure Meds, Exercise	Lisinopril, Ibuprofen	Penicillin	Blood Pressure: 140/90 mmHg	2023-11-20	Bill Smith (husband)
P003	Alice Brown	33	Female	Asthma	2023-02-11	Inhaler, Avoid Triggers	Albuterol	Pollen, Dust	Peak Flow: 80% of normal	2023-12-05	Charles Brown (brother)
P004	Michael Lee	52	Male	Coronary Artery Disease	2020-09-10	Cardiac Surgery, Medication	Aspirin, Statins	Shellfish	Cholesterol: 240 mg/dL	2023-12-01	Mary Lee (wife)
P005	Sarah White	29	Female	Migraine	2023-03-25	Pain relief, Rest	Sumatriptan, Ibuprofen	None	MRI: Normal	2023-12-12	David White (father)

2.2 Natural Language Processing

NLP in cloud healthcare has the potential to transform the sector by increasing healthcare accessible, improving patient outcomes, and providing healthcare professionals with sophisticated data-driven insights [5,22]. This comprises of the subsequent steps: spellchecking, capitalization, tokenization, and positional tagging and Named Entity Recognition (NER).

A. The spellcheck

Spellchecking is the automatic detection and correction of misspelled words in a given text. In healthcare, this is critical since medical texts (e.g., electronic health records, prescriptions, and clinical notes) contain specialist terminology that must be correctly written. Spelling errors in clinical paperwork might cause confusion or misinterpretation of medical terms, drug names, or therapies. A well configured spellchecking system can verify that patient records are correct. NLP tools must handle domain-specific terminology such as drug names, disease names, and medical procedures. NLP algorithms trained on medical lexicons are better able to detect and repair domain-specific spelling problems.

This query words are first checked for existence in the English dictionary. There is no need to check the spelling in this instance. If not, try coming up with some ideas by changing, replacing, or adding new letters to the term. From this list of possibilities, the term with the shortest processing distance is chosen.

Entry: What is blod cancr?

Output: What is blood cancer?

B. Capitalization

Capitalization is the process of accurately capitalizing specific words in text, notably the initial word of a phrase and proper nouns. To distinguish between common language and proper medical words, medical records and notes must follow capitalization norms (for example, ”cancer” could refer to a diagnosis, but ”cancer” could refer to general usage). Capitalization helps to arrange the content and improves clarity. Capitalization can also aid distinguish between common and proper nouns, allowing for more accurate identification of ailments, treatments, and specialists.

In general, if the query syntax is correct, using Natural Language Toolkit (NLTK) makes the query easier to understand. Words that are significant words or proper nouns should be capitalized for correct syntax. An online corpus of disease names is used to do this. Capitalize the ngram and alter the query, if the query ngram fits the name of the disease.

Entry: What is skin cancer?

Output: What is Skin Cancer?

Other NLP tasks, such as Part-of-Speech tagging, Notability Detection rely on tokenization to extract structured data from unstructured medical literature.

C. Tokenization and POS tagging

Tokenization is the act of dividing text into smaller parts, such as words or phrases, called tokens. It assists NLP systems in determining the structure of the text and preparing it for future analysis. Tokenization in medical texts enables the system to decipher clinical notes into specific words, keywords, or concepts. For example, "The patient was prescribed ibuprofen for fever" might be tokenized as ['the', 'patient', 'was', 'prescribed', 'ibuprofen', 'for', 'fever']. POS tagging is the process of determining the grammatical categories (or parts of speech) of certain tokens. For example, in the line "A doctor prescribed aspirin," "doctor" is a noun, but "prescribed" is a verb.

Tokenization tries to get a single entity from the query, and POS tagging attempts to tag each word with the similar tag.

Entry: Lung Cancer Tokenization is: ["what","is","skin","cancer"]

Posttagging: [("what","WH"),("is","VBZ"),("skin","NNP"),("cancer","NNP")]

D. Notability Detection

Notability detection in NLP is very crucial in healthcare since it entails finding significant entities such as medical conditions, treatments, pharmaceuticals, healthcare experts, and organizations with a major impact on health-related texts. The purpose is to discover essential entities that influence diagnosis, treatment, or healthcare decisions as mentioned in medical records, clinical studies, research articles, or even healthcare forums.

Noun entity recognition is done in three ways:

1) Use NLTK noun entity recognition
2) Use chunking dissimilar post tags organized and
3) Use noun if no noun entity is found

Entry: Whatislungcancer

Output: ["lung cancer"]

This work mainly uses the Fully Homomorphic Encryption (FHE) algorithm and Fernet encryption algorithm for two-layer security to protect EHHs.

2.3 Fully Homomorphic Encryption (FHE) Algorithm

The FHE algorithm is the standard and most trusted algorithm in cryptography, especially for text encryption, where keys are generated, and 128 bits are reduced to 16/32/64 bits for fast processing and balanced reading of blocks[23,24]. Rather than encrypting the source content, homomorphic encryption encodes previously encrypted data and returns the result in plain form. Ciphertext can be subjected to advanced mathematical processes while retaining its cryptographic properties.

The following stages were used to construct the FHE algorithm in the proposed work.

2.3.1 Encryption and Decryption

Steps for Encryption

Step 1: Key generation: Firstly set-of-spherical keys are gained from the Cipher key

Step 2: Loading of the array is implemented upon the plaintext / block array

Step 3: Next, the number one spherical – secrets introduced to the initialized array

Step 4: Next, country manipulation of 9 rounds is completed

Step 5: The 10^{th} array manipulation spherical is completed earlier acquisition of cipher text

Step 6: Finally, the array received is copied through the received ciphered - text / encrypted information.

Thus, the encryption is performed in which the 128bits collection is transformed as 16bytes for FHE because it plays nicely as 16 bytes.

Steps for Decryption

For the decryption the inverse capabilities are performed

Step 1: Initial decryption spherical is completed

Step 2: Nine-complete decryption rounds are followed

Step 3: Finally, the Xor Round key is completed to decrypt the cipher text

FHE Encryption Algorithm for encrypting Healthcare dataset

```
import random
import json

# Helper function to generate round keys from the cipher key

def generate_round_keys(cipher_key, num_rounds):
    round_keys = [cipher_key]
    for i in range(1, num_rounds):
        round_keys.append(round_keys[i-1] ^ random.randint(1, 255))  # Generate round key
            using XOR with randomness
    return round_keys


# Encryption function

def encrypt_healthcare_data(plaintext, cipher_key):
    rounds = 10  # Number of rounds as per the description
    block_size = len(plaintext)  # Size of the data being encrypted

    # Step 1: Key generation
    round_keys = generate_round_keys(cipher_key, rounds)

    # Step 2: Load the plaintext into an array and convert data into bytes/ASCII values
    block_array = [ord(c) for c in plaintext]

     # Step 3: Apply the first round key to the data
    block_array = [block_array[i] ^ round_keys[0] for i in range(block_size)]

    # Step 4: Perform 9 rounds of transformations (XOR with the round keys)
    for round_num in range(1, rounds - 1):
        block_array = [block_array[i] ^ round_keys[round_num] for i in range(block_size)]

    # Step 5: Apply the 10th (final) round key
    block_array = [block_array[i] ^ round_keys[rounds - 1] for i in range(block_size)]

    # Step 6: Convert the encrypted data into a final ciphertext (string)
    ciphertext = ''.join(chr(b) for b in block_array)  # Convert back to characters
    return ciphertext
```

FHE Decryption Algorithm for decrypting Healthcare dataset

```
# Decryption function
def decrypt_healthcare_data(ciphertext, cipher_key):
    rounds = 10  # Number of rounds as per the description
    block_size = len(ciphertext)  # Size of the encrypted data
    # Step 1: Key generation
    round_keys = generate_round_keys(cipher_key, rounds)
    # Step 2: Load the ciphertext into an array and convert each character back to its ASCII value
    block_array = [ord(c) for c in ciphertext]
     # Step 3: Perform the initial decryption round (reverse XOR with the last round key)
    block_array = [block_array[i] ^ round_keys[rounds - 1] for i in range(block_size)]
     # Step 4: Perform the 9 decryption rounds (reverse order of encryption rounds)
    for round_num in range(rounds - 2, -1, -1):
        block_array = [block_array[i] ^ round_keys[round_num] for i in range(block_size)]
    # Step 5: Convert the decrypted data back to plaintext
    plaintext = ''.join(chr(b) for b in block_array)
    return plaintext
```

```
# Healthcare Data
healthcare_data = {
   "patient_id": "12345",
   "name": "John Doe",
   "age": 35,
   "diagnosis": "Flu",
   "medications": ["Aspirin", "Cough Syrup"] }
plaintext = json.dumps(healthcare_data)
cipher_key = 123

# Encrypt the healthcare data

ciphertext = encrypt_healthcare_data(plaintext, cipher_key)

print("Encrypted Healthcare Data:", ciphertext)

# Decrypt the healthcare data back to original

decrypted_text = decrypt_healthcare_data(ciphertext, cipher_key)

print("Decrypted Healthcare Data:", decrypted_text)

# Convert the decrypted text back to a Python dictionary (healthcare data)

decrypted_data = json.loads(decrypted_text)

print("Decrypted Healthcare Data as JSON:", decrypted_data)
```

Thus, the set of rules is advanced and implemented upon the datasets to teach and check the accuracy of the advanced encoding and interpreting rounds in FHE. It is easier to construct a dependable security-based symmetric FHE since it has greater (longer) key lengths.

2.4 FERNET Encryption Algorithm

The Fernet algorithm is a symmetric encryption scheme that ensures data secrecy and integrity. It is part of Python's cryptography package and is designed to be easy and secure, making it appropriate for scenarios requiring straightforward and dependable encryption. Fernet uses plaintext for encryption and decryption in a manner similar to the FHE algorithm [25]. It offers rotation of keys produced by "MultiFernet".

2.4.1 Encryption and Decryption

The encryption key of Fernet follows three steps

Step1: Generates a key

Step2: Assign a key value to the selected variable

Step3: Convert plaintext to ciphertext

The plaintext is the output received from the FHE algorithm. To decrypt the encrypted text, Fernet performs the ciphertext-to-plaintext conversion as an inverse function, and the output is displayed as a "string" value of bytes.

Reasons for Adopting FHE and Fernet

Although there are various symmetric and asymmetric encryption and decryption algorithms, FHE is the most extensively used and standardized algorithm provides users with keys that are more secure than others [26,27]. The advantages using Fernet includes signature stamps, timestamps, random assignment of a "salt value" for security, key generation by a secure mechanism, and the adoption of secure algorithms such as Advanced Encryption Standard (AES), Data Encryption Standard (DES), Rivest, Shamir, Adleman (RSA), 3DES provides high-level encryption that makes key less data manipulation impossible.

Therefore, by combining FHE and Fernet, this work aims at advanced encryption and decryption of text as input to NLP-based encoder/decoder models, something that has not been attempted before in existing research

2.5 Get Ciphertext

EHH encryption techniques built on ciphertext recovery schemes can directly retrieve ciphertext information [7]. It ensures query privacy and improves search efficiency and also allows adding and multiplying search information without varying the equivalent plaintext. Figure 3 illustrates proposed symmetric EHH encryption, where the key is produced by a user-defined derived key generation function using the source code.

In EHH symmetric encryption (e-Healthsec), as shown in Figure 3, the key is generated using a user-defined derived key generation function using the source code.

”Generated Fernetkey=phJnZmXUMoe57l20jdwY7NbDmdsaCQCWAq8lc6dtEvQ=”

The key is a plaintext password with 128-bit encryption. In this layer, the trained first-layer dataset is Fernet-encoded and applied to the data input text. Once the data is encrypted, it is passed to EHH encryption whereas second level of encryption occurs as shown in below EHH Encryption algorithm e-Healthsec.

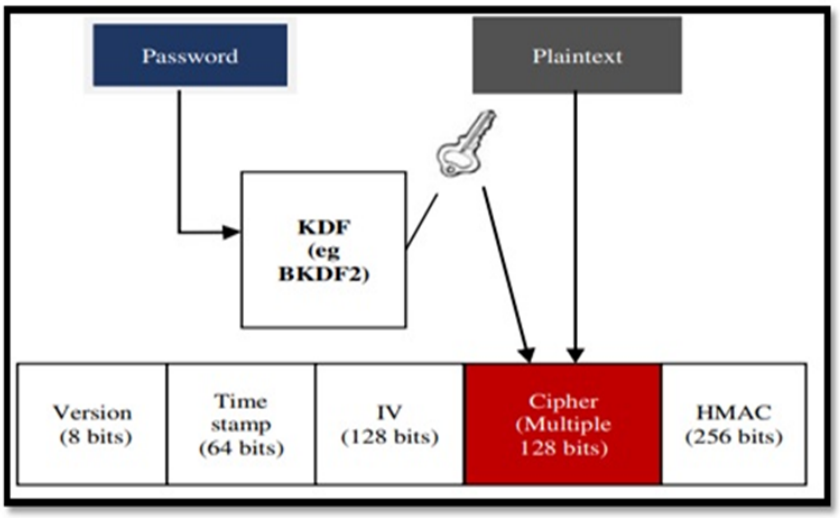

Fig 3. Proposed Symmetric EHH Encryption (e-Healthsec)

EHH Encryption Algorithm - e-Healthsec

```
#the Fernet module is crypto package

from cryptography.fernet import Fernet
#generate key
key=Fernet.generate_key()

#assign key value to variable
f=Fernet(key)

#Convert plaintext to ciphertext
token=f.encrypt(b"Welcome to EHH Encryption of  health data")

#show ciphertext
print(token)

#decrypt the ciphertext
d=f.decrypt(token)

# Convert bytes to string
print(d.decode())

#show plaintext
print(d)
```

2.6 AWS Cloud Implementation

The text file is collected after NLP then it is saved in local drive initially. The medical history data is stored as S3 bucket. The setup for the cloud storage of EHH model is demonstrated in Figure 4. It comprises of upload and downloads options. We have interfaces for both upload and download. When we click ”upload,” the file is encrypted, and the data is saved in a bucket as S3. This file will be in an encrypted format. Following this procedure, the content will be downloaded in encrypted format as seen in Figure 5.

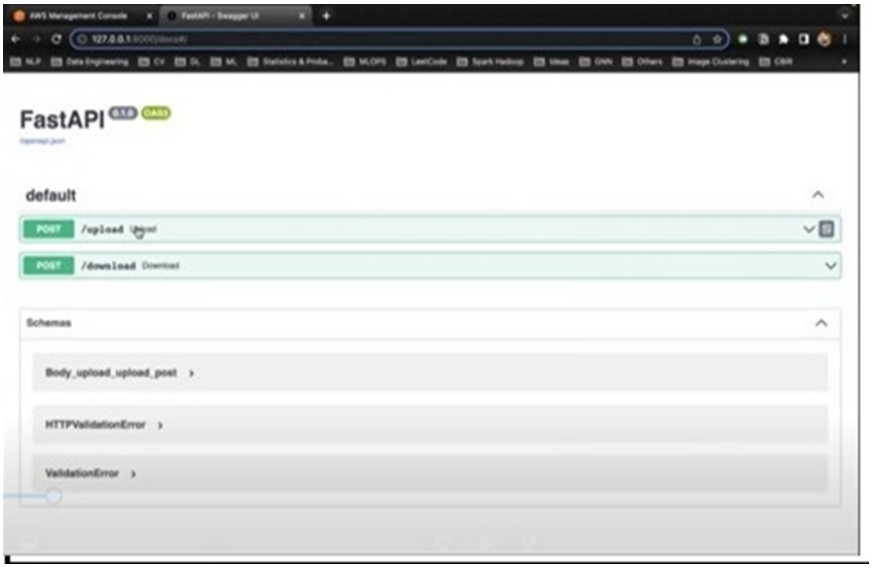

Fig 4. Local Cloud Setup API

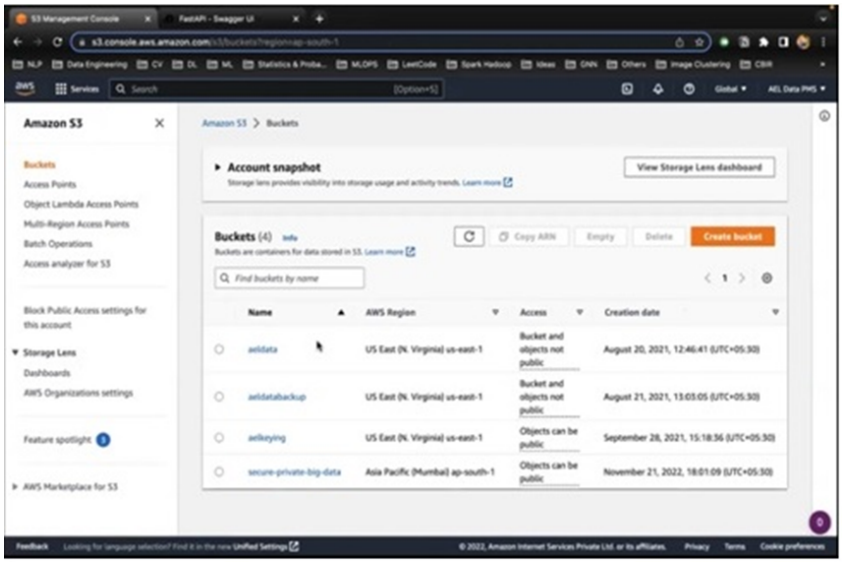

Fig 5. S3 files stored as encrypted

When using the download option, the same S3 file will be decrypted and stored in original format. Private key is used to encrypt all public information. Key is created and stored in local after loading every time.

3 Results and Discussion

The final proposed system comprises of the AWS cloud implementation, performance metrics such as level of protection, memory usage and bandwidth. Figure 6 displays the file's contents.

The home page for the cloud console is shown in Figure 7.

Figure 8 shows the S3 cloud bucket storage in Vsecure private bucket. It creates the secure private big data. The entire file will be uploaded directly in this bucket. For more security, isolated user is created. This user can have access to this S3 bucket. We have access and secret key through IAM user. The access key and secret key is shown in Figure 9.

```
#!pip install scispacy
#!pip install https://s3-us-west-2.amazonaws.com/ai2-s2-
scispacy/releases/v0.4.0/en_ner_bc5cdr_md-0.4.0.tar.gz
!pip install https://s3-us-west-2.amazonaws.com/ai2-s2-
scispacy/releases/v0.5.0/en_core_sci_sm-0.5.0.tar.gz


import pandas as pd
med_transcript = pd.read_csv("/content/sample_data/mtsamples.csv", index_col=
0)
med_transcript.info()
med_transcript.head()


<class 'pandas.core.frame.DataFrame'>
Int64Index: 4999 entries, 0 to 4998
Data columns (total 5 columns):
#   Column             Non-Null Count  Dtype
---  ------             --------------  -----
0   description        4999 non-null   object
1   medical_specialty  4999 non-null   object
2   sample_name        4999 non-null   object
3   transcription      4966 non-null   object
4   keywords           3931 non-null   object
dtypes: object(5)
memory usage: 234.3+ KB

med_transcript.dropna(subset=['transcription'], inplace=True)
med_transcript_small = med_transcript.sample(n=100, replace=False, random_sta
te=42)
med_transcript_small.info()
med_transcript_small.head()


# Let's take one transcription to see how we can work with NER:
# Named entity recognition (NER) is a subtask of natural language processing
used to identify and
# classify named entities mentioned in unstructured text into pre-
defined categories.
sample_transcription = med_transcript_small['transcription'].iloc[0]
print(sample_transcription[:1000]) # prints just the first 1000 characters

HISTORY OF PRESENT ILLNESS:,  The patient is well known to me for a history
of iron-deficiency anemia due to chronic blood loss from colitis.  We
corrected her hematocrit last year with intravenous (IV) iron.  Ultimately,
she had a total proctocolectomy done on 03/14/2007 to treat her colitis.  Her
course has been very complicated since then with needing multiple surgeries
for removal of hematoma.  This is partly because she was on anticoagulation
for a right arm deep venous thrombosis (DVT) she had early this year,
complicated by septic phlebitis.,Chart was reviewed, and I will not reiterate
her complex history.,I am asked to see the patient again because of concerns
```

Fig 6. Content of the EHH File

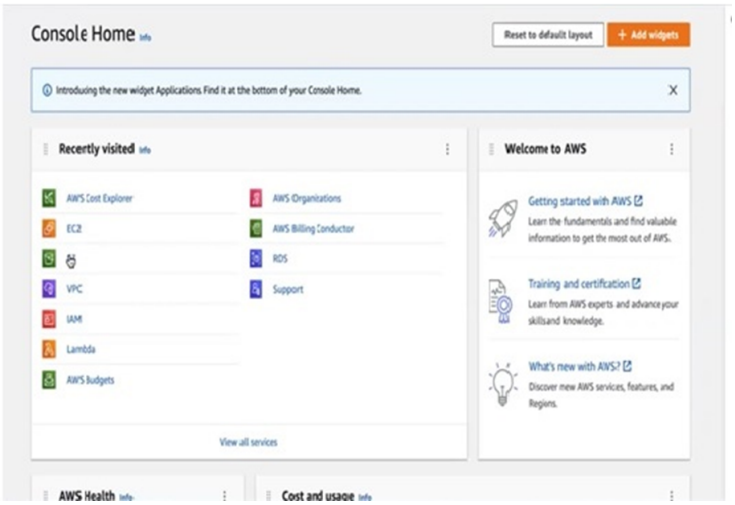

Fig 7. Cloud Console Home Page

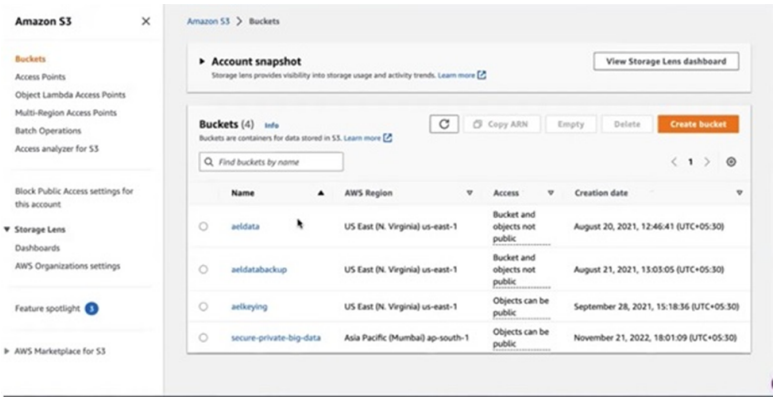

Fig 8. Creative VSecure Cloud S3 Bucket Storage

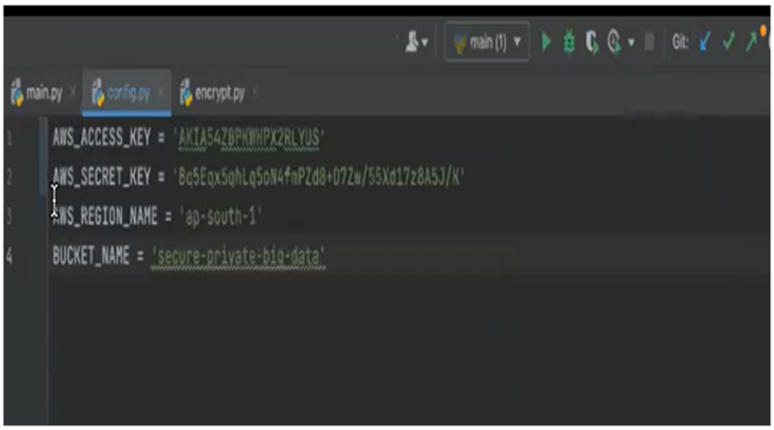

Fig 9. Access Key and Secret Key

Figure 10 illustrates the summary of the limited edition of S3 bucket. Through IAM, we created users, roles, group assign and tag the setup individually.

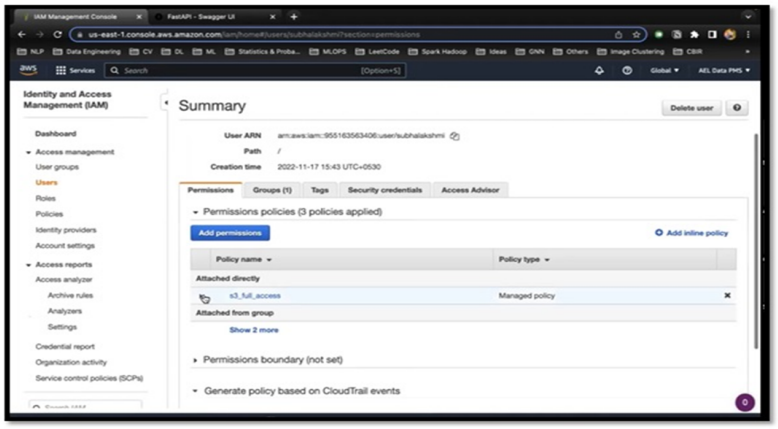

Fig 10. Summary of the S3 Bucket

Figure 11 shows the memory usage of different encryption algorithms for FHE and proposed EHH encryption algorithm(e-Healthsec) using PYFHEL. The unit measured in terms in kilobytes (KB).

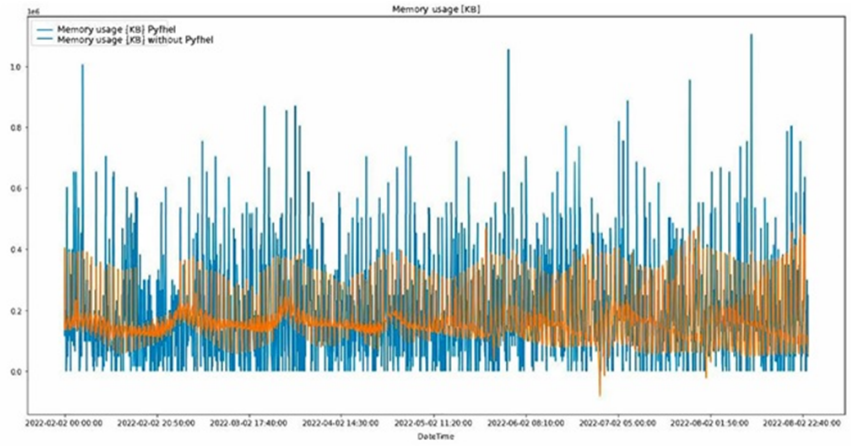

Fig 11. Memory Usage of Proposed Framework

Figure 12 shows the throughput comparison between DES, AES, RSA, 3DES, FHE, Fernet and EHH encryption algorithm(e-Healthsec). The time taken is measured in terms of number of bytes /sec.

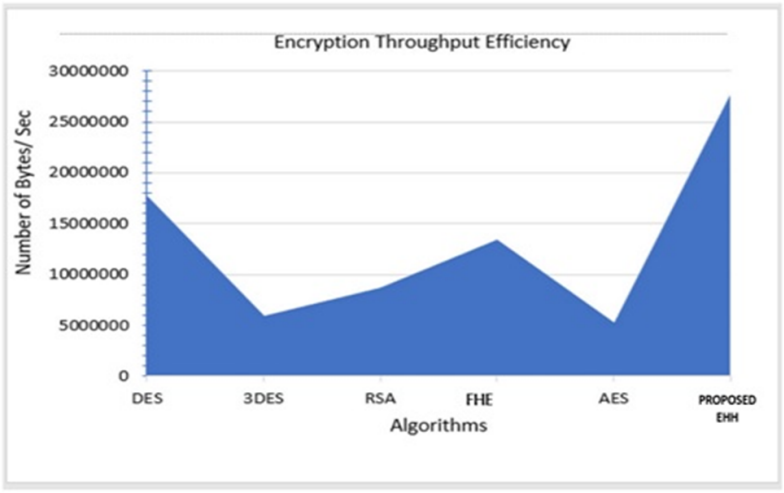

Fig 12. Throughput Comparison of Encryption algorithms

The file is running at different size susing AES, DES, FHE, RSA, 3DES and proposed EHH encryption algorithm (e-Healthsec). An overall examination of these processes is evaluated based on parameters such as execution time, memory relative to implementation, and throughput. Python is cast-off to create a conventional comparison model EHH. The proposed encryption strategy aims to determine which of the aforementioned encryption techniques is the fastest and most secure. This allows users to modify the encryption algorithm that best suits the type of information being encrypted. The results show that the proposed EHH system(e-Healthsec) of algorithms raises the productivity of encryption algorithms by firmly encrypting data in short period of time. Time-Based Analysis and Evaluation of Encryption and Decryption analysis shows that e-Healthsec is faster than all other algorithms, and the system outperforms all other algorithms. This works numerous times faster than Key generation must be finished before encrypting data. This method is implemented with the cooperation of the patients and the cloud service provider. Compared to prior encryption processes, the EHH encryption (e-Healthsec) process takes 10-15% less time to encrypt files. The inverse of encryption time is known as decryption time. This is the premeditated time it takes for the decryption algorithm to create the original text from the encrypted text. By analyzing the decoding time, the hybrid system of algorithms performs better than all other algorithms. Figure 13 shows the comparison of encryption times for AES, DES, FHE, and Proposed EHH encryption algorithm (e-Healthsec).

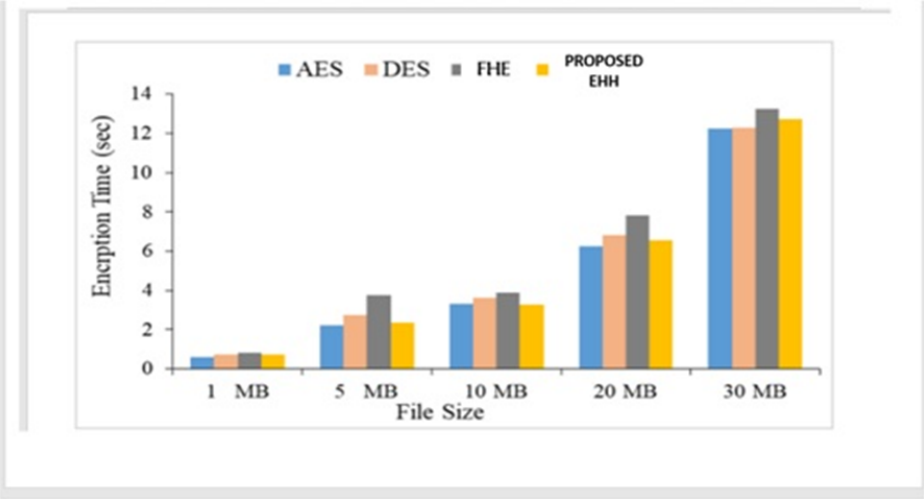

Fig 13. Comparison of Encryption time of AES, DES, FHE, and Proposed EHH Algorithm (e-Healthsec)

4 Conclusion

As the demands of healthcare big data grow, sustaining the integrity of the big data, cloud and its users turn out to be a top significance. Despite the fact that there are several cloud security measures accessible, multi-layer encryption methods are required to assure the cloud storage security of medical data histories.

With the increased usage of big data to store large amounts of data, security and confidentiality have become key challenges. This paper suggests developing an encryption model that combines the effect of NLP and text mining with the EHH data encryption algorithm (e-Healthsec) to protect the part of data in documents that need protection. This approach encrypts attributes into the text itself, which speeds up its processing. A comparison of the performance of the AES, DES, Fernet, FHE, and suggested EHH encryption algorithm (e-Healthsec) reveals that the new technique surpasses existing methods in terms of memory usage, throughput, and encryption time. Compared to previous efforts, this technology performs EHH encryption and decryption in 10-15% of lesser time. Healthcare oorganizations can provide strong data security by implementing EHH encryption framework – e-Healthsec which uses the Fernet encryption for data confidentiality and homomorphic encryption for secure computing. This combination is particularly useful in privacy-sensitive environments, such as healthcare, where confidentiality and secure processing of sensitive data are crucial. The key advantage of this proposed system is that it saves confidential medical histories on the cloud, which will be useful in an emergency.

Future Work

In the future, the industry based on cloud computing will be greatly hindered, and the challenge remains that users will lose data and information stored in the cloud. According to this research's findings, cloud design and taxonomies make it simpler for outsiders to access the cloud and steal sensitive data; as a result, the author believes that modifying the approach to architecture is the best way to address this problem. Likewise, users should choose remote access with two levels of encryption to safeguard their data.

References

1) Kedi WE, Ejimuda C, Ajegbile MD. Cloud computing in healthcare: A comprehensive review of data storage and analysis solutions. *World Journal of Advanced Engineering Technology and Sciences*. 2024;12(02):290–298. Available from: https://doi.org/10.30574/wjaets.2024.12.2.0291.
2) Javid T, Faris M, Beenish H, Fahad M. Cybersecurity and Data Privacy in the Cloudlet for Preliminary Healthcare Big Data Analytics. In: 2020 International Conference on Computing and Information Technology (ICCIT-1441). 2020. Available from: https://doi.org/10.1109/ICCIT-144147971.2020.9213712.
3) Tang Z. A Preliminary Study on Data Security Technology in Big Data Cloud Computing Environment. In: 2020 International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE). IEEE. 2021. Available from: https://doi.org/10.1109/ICBASE51474.2020.00013.
4) Wang F, Wang H, Xue L. Research on Data Security in Big Data Cloud Computing Environment. In: 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC);vol. 2021. IEEE. 2021;p. 1446–1450. Available from: https://doi.org/10.1109/IAEAC50856.2021.9391025.

5) Basha US, Gupta SK, Alawad W, Kim S, Bharany S. Fortifying Healthcare Data Security in the Cloud: A Comprehensive Examination of the EPM-KEA Encryption Protocol. *Computers, Materials & Continua*. 2024;79(2):3397–3416. Available from: https://doi.org/10.32604/cmc.2024.046265.
6) Wang Z, Wang N, Su X, Ge S. An empirical study on business analytics affordances enhancing the management of cloud computing data security. *International Journal of Information Management*. 2020;50:387–394. Available from: https://doi.org/10.1016/j.ijinfomgt.2019.09.002.
7) Srivenkateswaran C, Rani AJM, Kumaran RS, Raja RV. Securing healthcare data: A federated learning framework with hybrid encryption in cluster environments. *Technology and Health Care*. 2024;(0):1–26. Available from: https://dx.doi.org/10.1177/09287329241291397.
8) Ganiga R, Pai RM, Pai MMM, Sinha RK. Security framework for cloud based electronic health record (EHR) system. *International Journal of Electrical and Computer Engineering (IJECE)*. 2020;10(01):455–466. Available from: http://doi.org/10.11591/ijece.v10i1.pp455-466.
9) Dubey P. AWS Cloud for Remote Medical Monitoring. *CVRU conference*. 2023. Available from: https://doi.org/10.6084/m9.figshare.24586518.
10) Neela S, Neyyala Y, Pendem V, Peryala K, Kumar V, Cloud V. Computing Based Learning Web Application through Amazon Web Services. In: 7th International Conference on Advanced Computing and Communication Systems (ICACCS). IEEE. 2021;p. 472–475. Available from: https://doi.org/10.1109/ICACCS51430.2021.9441974.
11) Singh T. The effect of Amazon Web Services (AWS) on Cloud-Computing. *International Journal of Engineering Research & Technology (IJERT)*. 2021;10(11):480–482. Available from: https://doi.org/10.17577/IJERTV10IS110188.
12) Mali A, Bendale S. Cloud Security with AWS. *International Research Journal of Modernization in Engineering Technology and Science*. 2022;4(11):121–125. Available from: https://www.doi.org/10.56726/IRJMETS30997.
13) Naidu PR, Gowda VD, Mali US, Mallikarjun S, Srinivas D, Kawale SR. Cloud-Based Multi-Layer Security Framework for Protecting E-Health Records. *2023 International Conference on Artificial Intelligence for Innovations in Healthcare Industries (ICAIIHI) 2023; 1-7*. 2023;p. 1–7. Available from: https://www.doi.org/10.1109/ICAIIHI57871.2023.1048978.
14) Hossaine R, Higgins N, Soar J, Barua PD, Pisani AR, Turner K. Natural Language Processing in Electronic Health Records in relation to healthcare decision-making: A systematic review. *Computers in Biology and Medicine*. 2023;155. Available from: https://doi.org/10.1016/j.compbiomed.2023.106649.
15) Gong Y, Chang X, Misic J, Wang J, Zhu H. Practical solutions in fully homomorphic encryption: a survey analyzing existing acceleration methods. *Cyber Security-Springer Open*. 2024;5:1–23. Available from: https://doi.org/10.1186/s42400-023-00187-4.
16) Dourado LDS, Miranda RS, de Araujo APF, Ishikawa EE. Performance Evaluation of Big Data Applications in Cloud Providers. In: 15th Iberian Conference on Information Systems and Technologies (CISTI). IEEE. 2020. Available from: https://doi.org/10.23919/CISTI49556.2020.9140855.
17) Venkatesh BH, Saiap, Reddy MR, Fathimabi SK. Cloud based Personal Health Record Management System and Medical Recommender System. In: 7th International Conference on Communication and Electronics Systems (ICCES). IEEE. 2022;p. 1744–1749. Available from: https://doi.org/10.1109/ICCES54183.2022.9835988.
18) Nguyen TN, Le TTT. Authentication and Encryption algorithms for data security in Cloud computing: A Comprehensive review. In: Proceedings of the 2021 Sixth International Conference on Research in Intelligent and Computing;vol. 27. 2021;p. 57–63. Available from: http://dx.doi.org/10.15439/2021R7.
19) Ali S. A Comprehensive Study on Security and Privacy of E-Health Cloud-Based System. In: Silhavy, R, Silhavy, P, editors. Cybernetics and Control Theory in Systems. CSOC 2024;vol. 1119 of Lecture Notes in Networks and Systems. Cham. Springer. 2024;p. 1–31. Available from: https://doi.org/10.1007/978-3-031-70300-3_1.
20) Kaur J, Rani R, Kalra N. Healthcare Data Security and Privacy Protection Framework Based on Dual Channel Blockchain. *Transactions on Emerging Telecommunications Technologies*. 2025;36(1). Available from: https://doi.org/10.1002/ett.70049.
21) Wenhua Z, Hasan MK, Jailani NB, Islam S, Safie N, Albarakati HM, et al. A lightweight security model for ensuring patient privacy and confidentiality in telehealth applications. *Computers in Human Behavior*. 2024;153. Available from: https://dx.doi.org/10.1016/j.chb.2024.108134.
22) Zhou B, Yang G, Shi Z, Shaodan M. Natural Language Processing for Smart Healthcare. *IEEE Reviews in Biomedical Engineering* . 2022;17:4–18. Available from: https://doi.org/10.1109/RBME.2022.3210270.
23) Kamble A, Jiet MM, Puri C. Homomorphic Encryption and its Applications in Multi-Cloud Security. In: 2024 International Conference on Inventive Computation Technologies (ICICT). IEEE. 2024;p. 1493–1499. Available from: https://doi.org/10.1109/ICICT60155.2024.10544773.
24) Alabdulatif A, Khalil I, Yi X. Towards secure big data analytic for cloud-enabled applications with fully homomorphic encryption. *Journal of Parallel and Distributed Computing*. 2020;137:192–204. Available from: https://doi.org/10.1016/j.jpdc.2019.10.008.
25) Anon P, Tyagi SS. Enhancing Security of Cloud Data through Encryption with AES and Fernet Algorithm through Convolutional-Neural-Networks (CNN). *International Journal of Computer Networks and Applications*. 2021;8(4):288–299. Available from: https://doi.org/10.22247/ijcna/2021/209697.
26) Ibarrondo A, Viand A. Pyfhel: PYthonForHomomorphic Encryption Libraries. In: Proceedings of the 9th on Workshop on Encrypted Computing & Applied Homomorphic Cryptography. 2021;p. 11–16. Available from: https://doi.org/10.1145/3474366.3486923.
27) Mittal S, Ramkumar KR. Research perspectives on fully homomorphic encryption models for cloud sector. *Journal of Computer Security*. 2021;29(2):135–160. Available from: https://dx.doi.org/10.3233/jcs-200071.