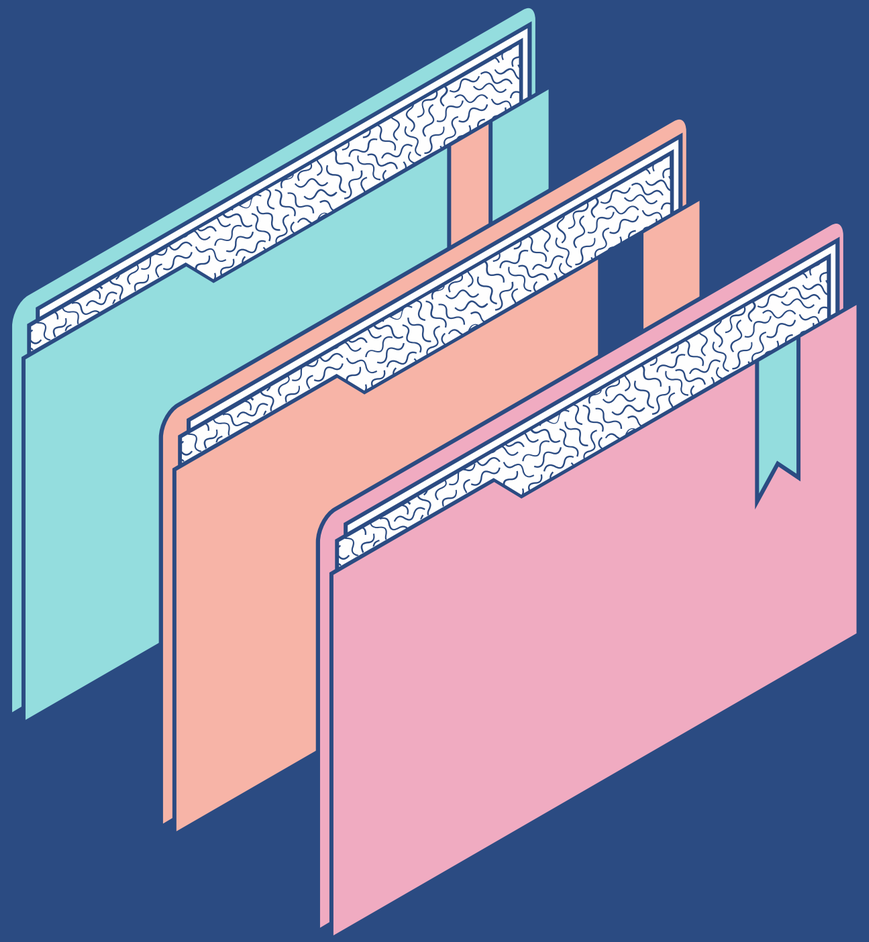
Homomorphic Encryption on Network traffic data for Neural Networks

UTS Computer Science Studio 2

Nam Khanh Cao, Diyon Ratnayaka, Jayce, Aurora Wu, Brendan Budniak



Agenda

- Thesis Statement
- Background and related work
- Motivation
- Problem Formulation
- Problem Solution
- Experiments
- Conclusion and Future Direction

Secures privacy, maintains encrypted data operations.

“Examine the efficacy of integrating the Paillier cryptosystem's homomorphic encryption into neural networks as a means to enhance the security of network traffic data.”

Background and Related Work

A COMPREHENSIVE OVERVIEW OF RESEARCH TOPIC

1 .Public-Key Cryptosystems Based on Composite Degree Residuosity Classes

Pascal Pallier

2. CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy

Nathan Dowlin, Ran Gilad-Bachrach, Kim Laine, Kristin Lauter, Michael Naehrig, John Wernsing

3. Privacy-Preserving Convolutional Neural Networks Using Homomorphic Encryption

Tatjana Wingarz, Marta Gomez-Barrero, Christoph Busch and Mathias Fischer

Public-Key Cryptosystems Based on Composite Degree Residuosity Classes

BY PASCAL PALLIER

Paillier Encryption Scheme; known for it’s additive homomorphism properties

Paillier introduced the theoretical framework, including the mathematical operations and properties that provide the foundation for the security of the encryption technique.

What is additive Homomorphism ?

Given only the ciphertexts corresponding to plaintexts $m1$ and $m2$, along with the same public key, anyone can compute the ciphertext corresponding to the sum of $m1$m 1and $m2$m 2 without needing to decrypt the ciphertexts.

CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy

BY NATHAN DOWLIN, RAN GILAD-BACHRACH, KIM LAINE, KRISTIN LAUTER, MICHAEL NAEHRIG, JOHN WERNSING

Combines homomorphic encryption techniques with neural networks

The core idea is data remain encrypted throughout the entire machine learning process, including training phases, to address concerns regarding privacy and security in machine learning applications.

Challenges and limitations

CryptoNets: the computational overhead with homomorphic encryption as it adds extra computational complexity.

Privacy-Preserving Convolutional Neural Networks Using Homomorphic Encryption

BY TATJANA WINGARZ, MARTA GOMEZ-BARRERO, CHRISTOPH BUSCH AND MATHIAS FISCHER

Privacy-preserving CNNs using homomorphic encryption

The network, implemented via SEAL-Python, a Python wrapper, achieved a precision in training that led to a final scaling of encryption numbers up to 2^234.

High Resource Consumption

Resource consumption of the encrypted network is significantly higher than the baseline system, '***we can approximate the size of a single ciphertext by 2*Nlog2(q) bits***' as stated by the author.

PROBLEM FORMULATION

Core Problem Identification

- Vulnerability of network traffic data during analysis
- Traditional encryption methods fail during processing, exposing data to breaches
- Machine learning algorithms require access to unencrypted data, increasing security risks

Challenges with Current Encryption Methods

- Security Vulnerabilities
  - Data must be decrypted for processing, increasing breach risk
- Performance Limitations
  - Encryption hinders ML algorithm performance, limiting data computations
- Data Encoding Issues
  - Mismatch between ML data types and encryption requirements
- Computational Overhead
  - High computational demand slows analysis and decision-making

MOTIVATION

- **Secure data processing** - Enables computations on encrypted data without exposing it.
- **Enhances privacy and security** - Protects data from unauthorized access during analysis.
- **Crucial for machine learning** - Allows secure use of data in distributed environments.

"Validate the effectiveness of Paillier encryption in processing encrypted network traffic data within machine learning frameworks"

Proposed Solutions

Solutions

Multi-processing for faster encryption

Using multiple processes for our CPU to perform encryption in parallel with different parts of the input data

---

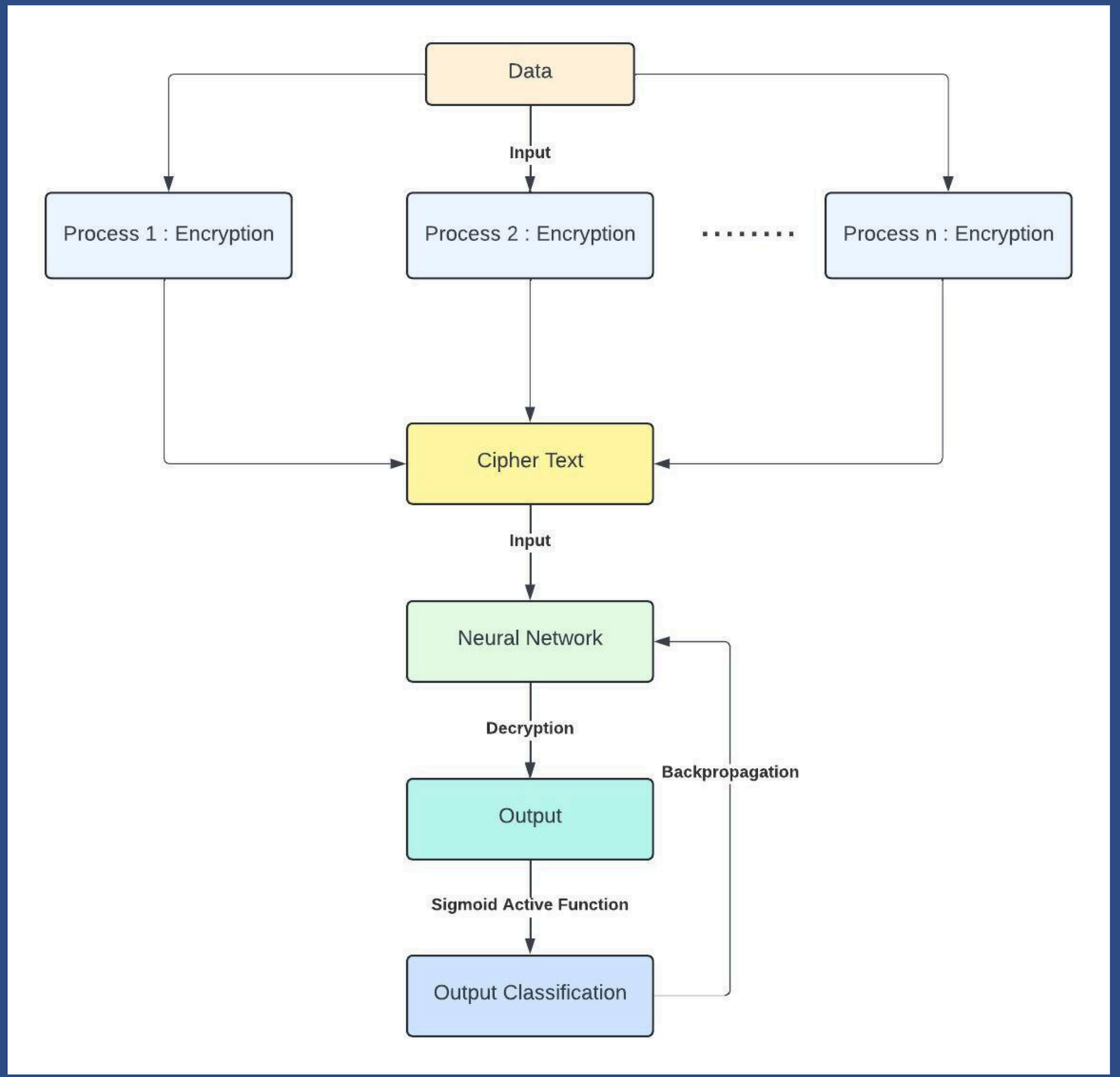
Efficient usage of numpy arrays

Usage of numpy arrays in backpropagation instead of list comprehension by decrypting output between layers before backpropagation.

Encoding Wrapper

Implementing a Encoding wrapper for efficient storage of encoded numbers for encryption.

Solutions


Data
Input
Process 1 : Encryption
Process 2 : Encryption
.....
Process n : Encryption
Cipher Text
Input
Neural Network
Decryption
Backpropagation
Output
Sigmoid Active Function
Output Classification

Encoding Wrapper

Wrap around scalar with a large encoded scalar

Decode after decryption

Capable of converting floating point numbers into integers to satisfy a requirement for Paillier Scheme

Avoid overflows by having an exponent attribute

Multiprocessing encryption

Select minimums and maximums

Set up a pool of processes

Assign each process a certain range in between input data

Concatanate the results after processes finish encrytion

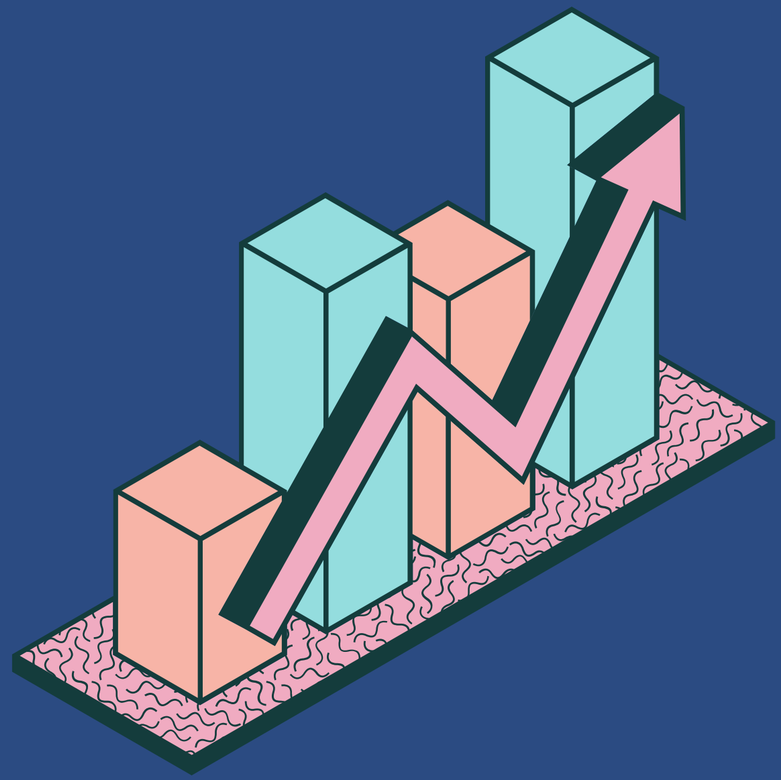
Efficient usage of numpy

Convert layer by layer results to numpy by decrypting

Can use activation functions

Backpropagation uses numpy for matrix multiplication instead of relying on list comprehension

Experiment



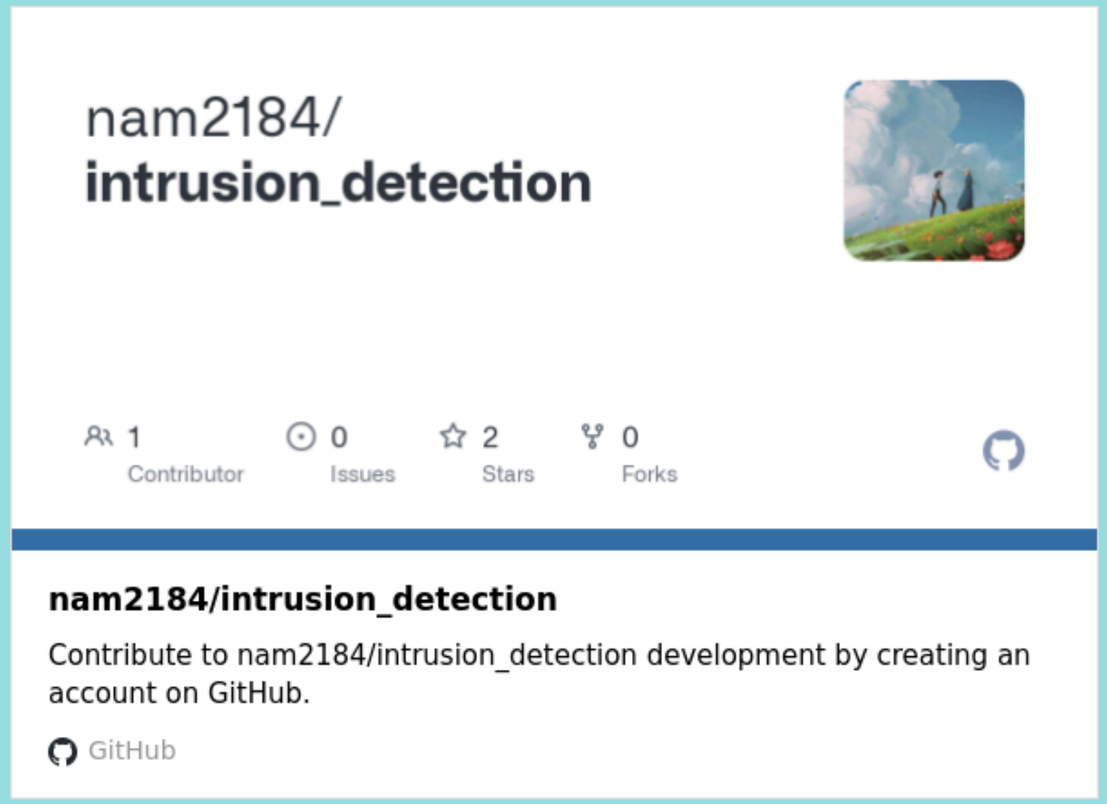
In this section

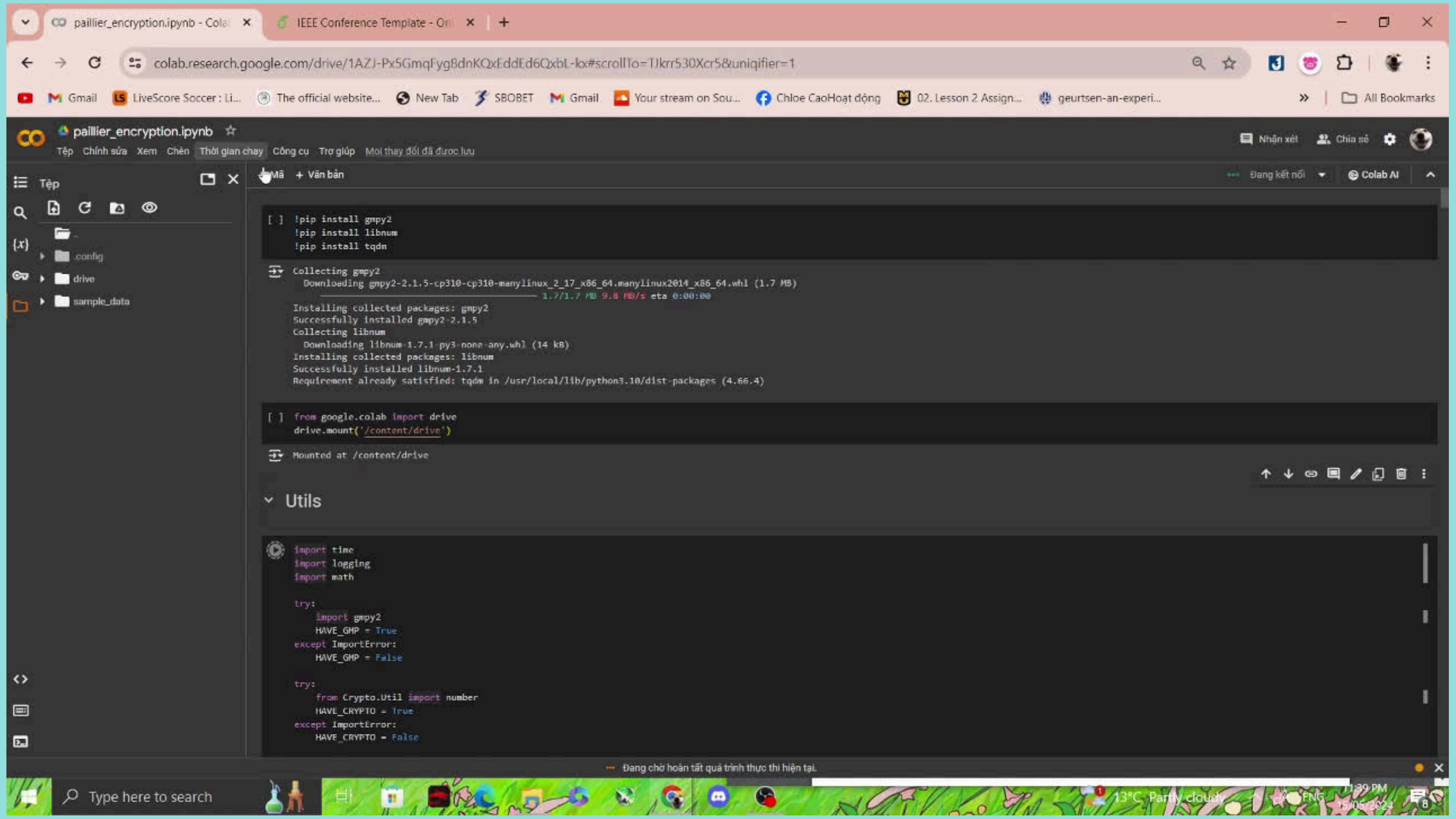
We proposed optimize our ML model throughout multiple experiments for KDD-Cup 1999 dataset.

- MLP
- RNN

Additional Traditional Model only work for unencrypted data

- SVM
- KNN
- Gradient Boosting




paillier_encryption.ipynb - Cola
IEEE Conference Template - On
colab.research.google.com/drive/1AZJ-Px5GmqFyg8dnKQxEddEd6QxbL-kx#scrollTo=TJkrr530Xcr5&uniqifier=1
Gmail
LiveScore Soccer : Li...
The official website...
New Tab
SBOBET
Gmail
Your stream on Sou...
Chloe CaoHoạt động
02. Lesson 2 Assign...
geurtsen-an-experi...
All Bookmarks
paillier_encryption.ipynb
Tệp Chỉnh sửa Xem Chèn Thời gian chạy Công cụ Trợ giúp Mọi thay đổi đã được lưu
Nhận xét
Chia sẻ
Tệp
Mã + Văn bản
Đang kết nối
Colab AI
.config
drive
sample_data
!pip install gmpy2
!pip install libnum
!pip install tqdm
Collecting gmpy2
Downloading gmpy2-2.1.5-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.7 MB)
1.7/1.7 MB 9.8 MB/s eta 0:00:00
Installing collected packages: gmpy2
Successfully installed gmpy2-2.1.5
Collecting libnum
Downloading libnum-1.7.1-py3-none-any.whl (14 kB)
Installing collected packages: libnum
Successfully installed libnum-1.7.1
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (4.66.4)
from google.colab import drive
drive.mount('/content/drive')
Mounted at /content/drive
Utils
import time
import logging
import math
try:
import gmpy2
HAVE_GMP = True
except ImportError:
HAVE_GMP = False
try:
from Crypto.Util import number
HAVE_CRYPTO = True
except ImportError:
HAVE_CRYPTO = False
Đang chờ hoàn tất quá trình thực thi hiện tại.
Type here to search
13°C Partly cloudy
ENG
1:39 PM
15/05/2024

ENCRYPTION RATES

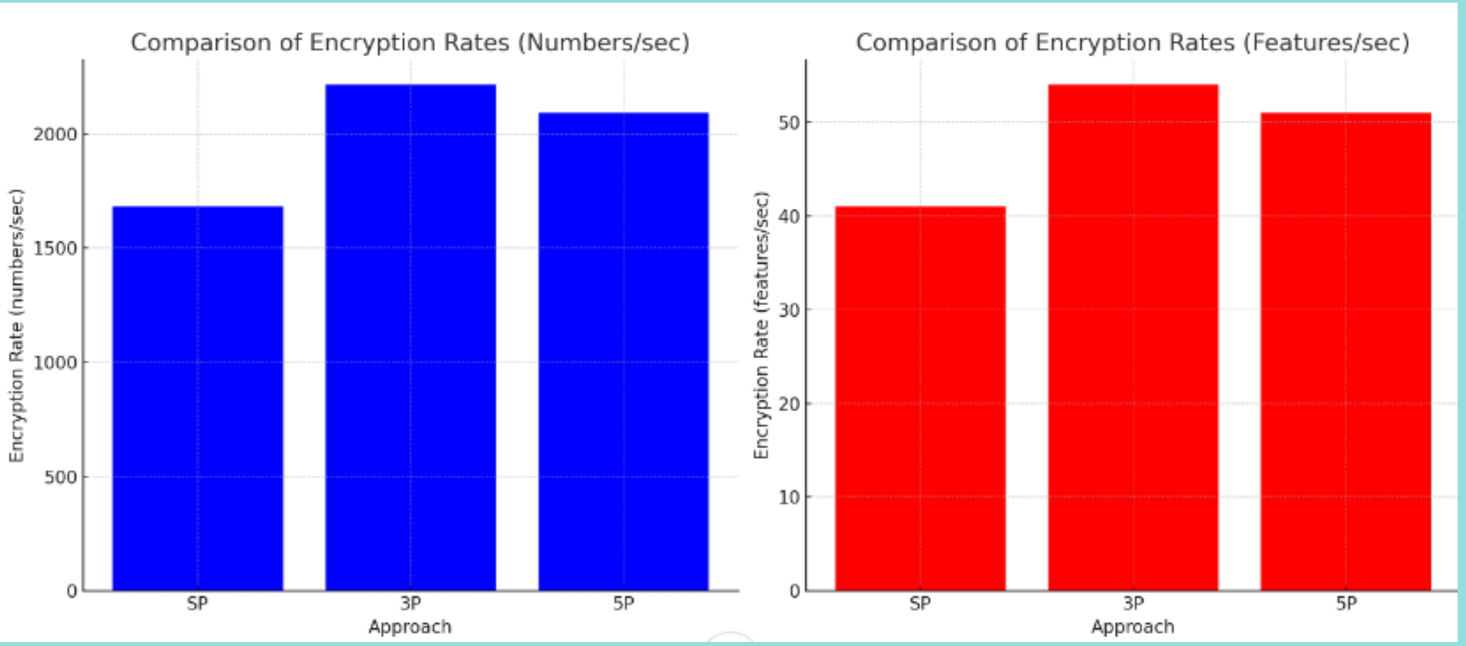


TABLE II
COMPARISON OF ENCRYPTION RATES (NUMBERS/SEC)

Approach	Processes	Encryption Rate (Numbers/sec)
Single Process	1	1681
Multiprocess (3 processes)	3	2214
Multiprocess (5 processes)	5	2091

TABLE III
COMPARISON OF ENCRYPTION RATES (FEATURES/SEC)

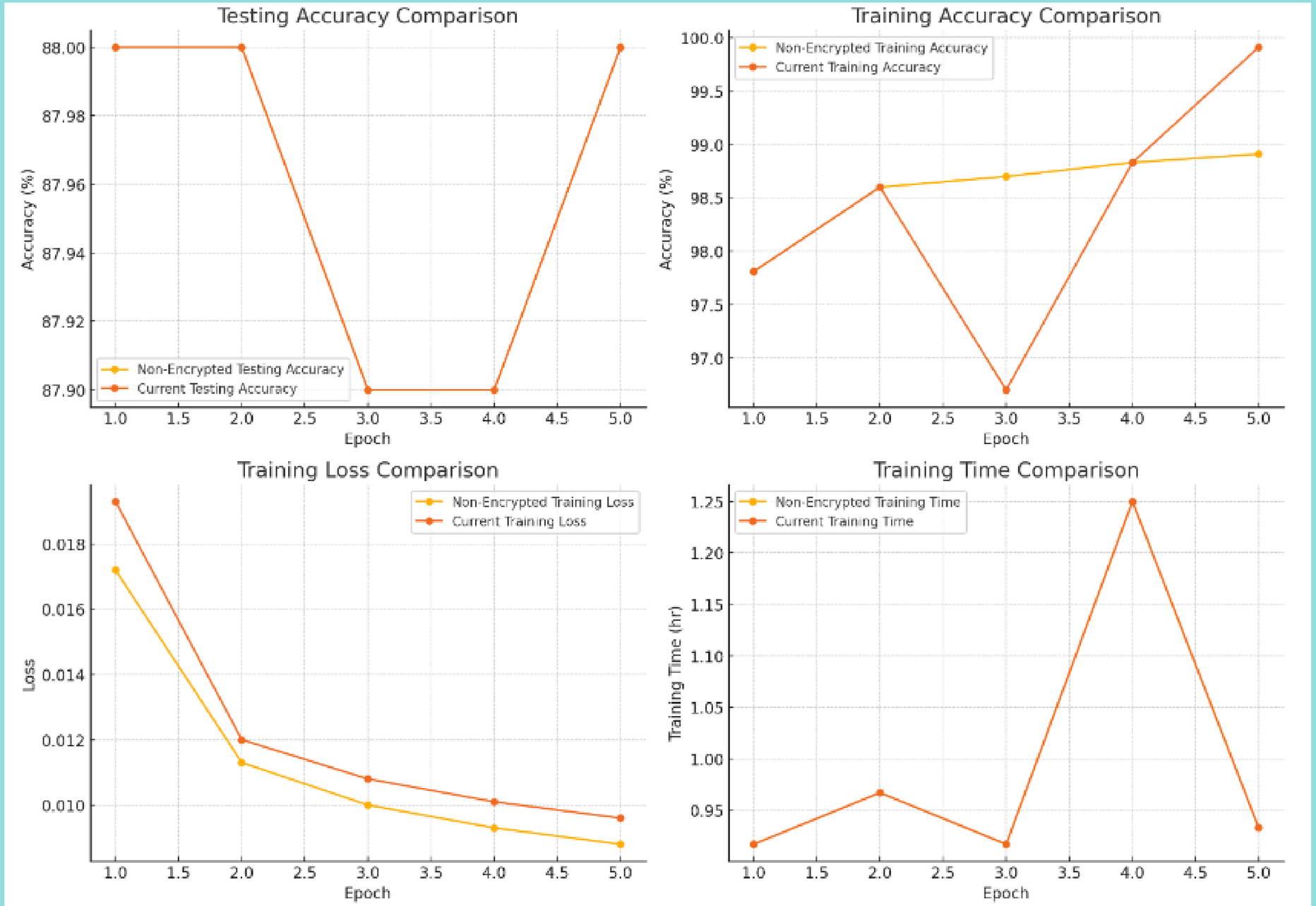
Approach	Processes	Encryption Rate (features/sec)
Single Process	1	41
Multiprocess (3 processes)	3	54
Multiprocess (5 processes)	5	51

[a]Depends on hardware specifications.

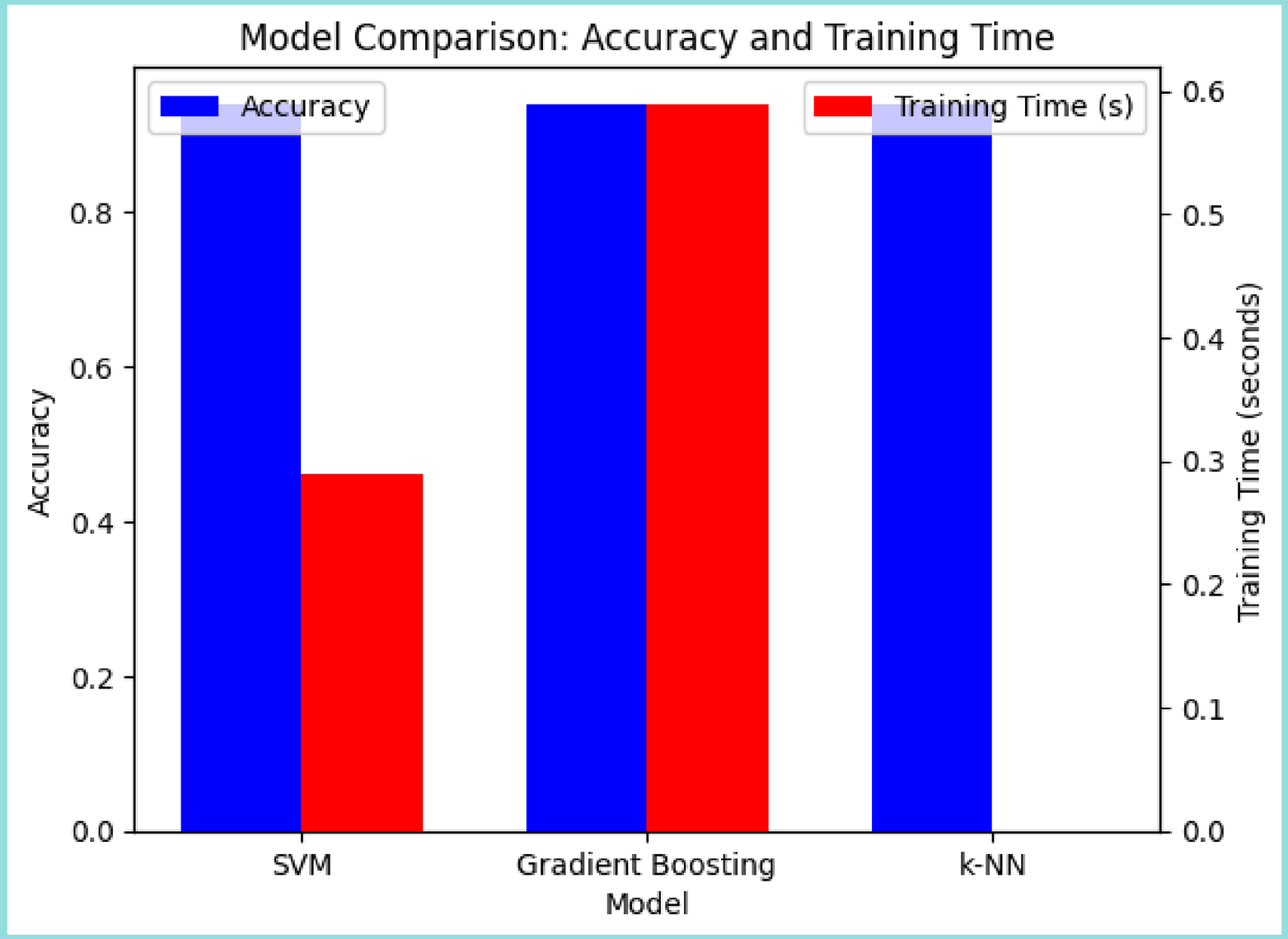
Input size	Hidden Size	Output size	Learning rate	Training sample size	Testing sample size
41	41	5	0.01	10000	1000

Normal	u2r	dos	r2l	probe
Normal packets	User to Root attacks (Privilege escalation)	Denial of Service attacks	Remote to local attacks	Probing or Scanning the network traffic (e.g. Port scans)

RNN INITIAL AND CURRENT EXP

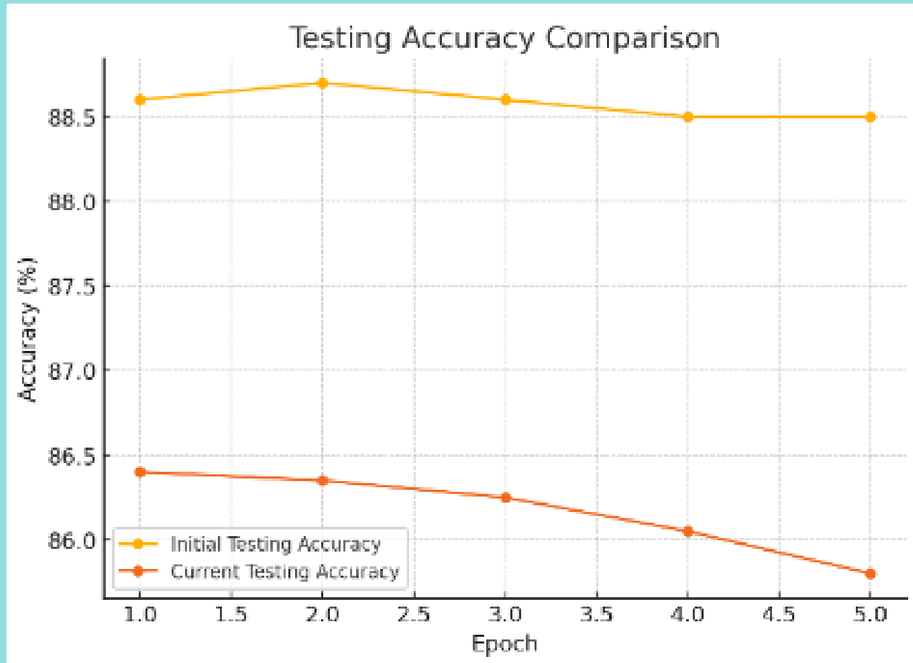


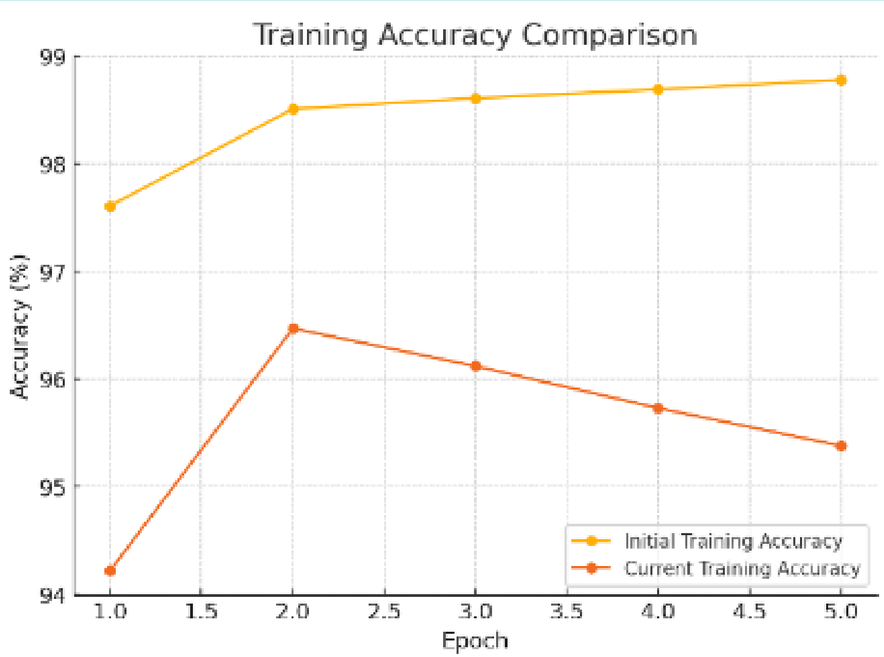
SVM GRB KNN WITH UNENCRYPTED DATA

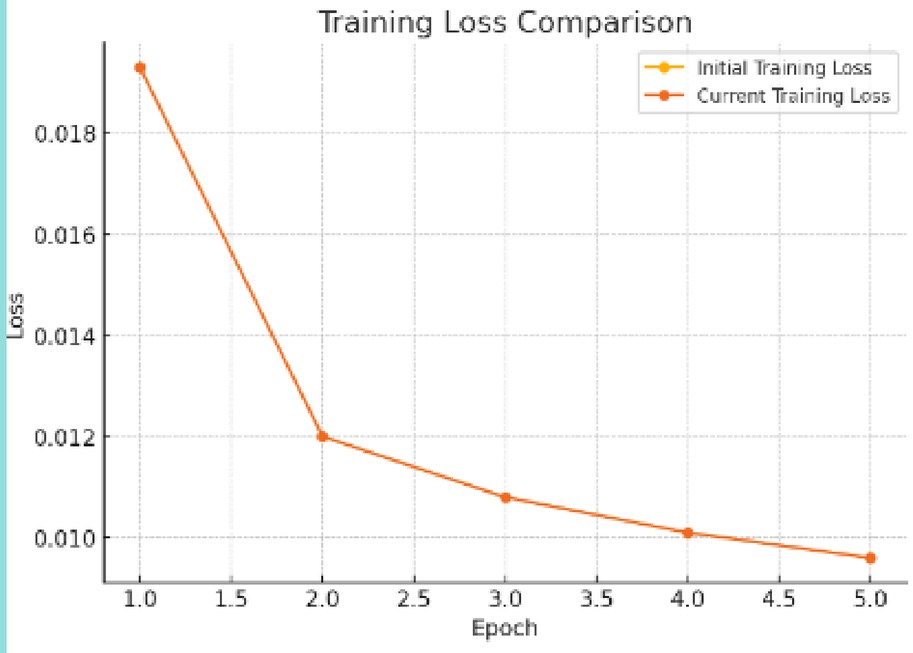


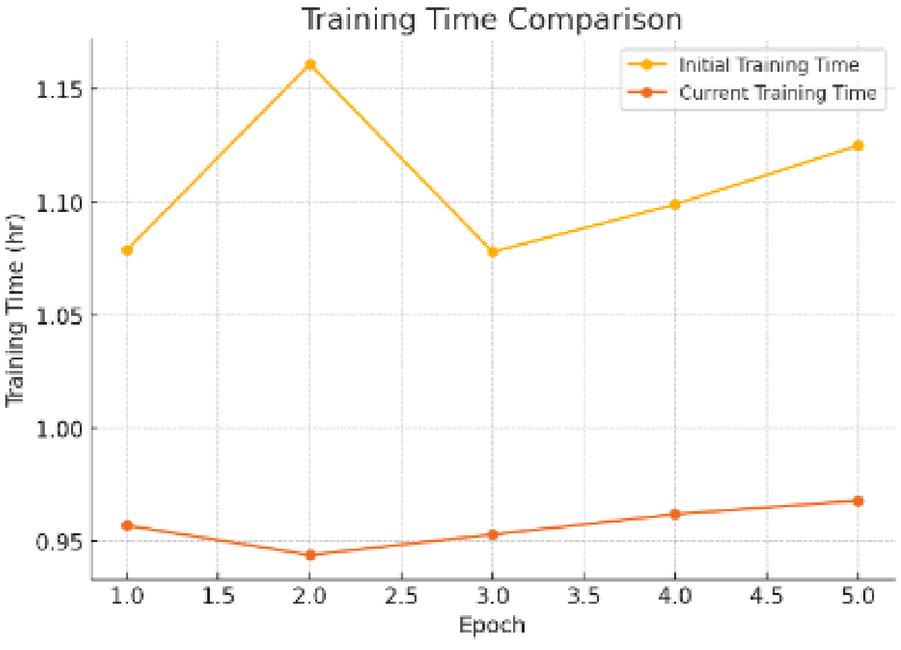
Metric	SVM	Gradient Boosting	k-NN
ACC%	0.94	0.94	0.94
Training Time	0.29	0.59	0.00

MLP INITIAL AND CURRENT EXP

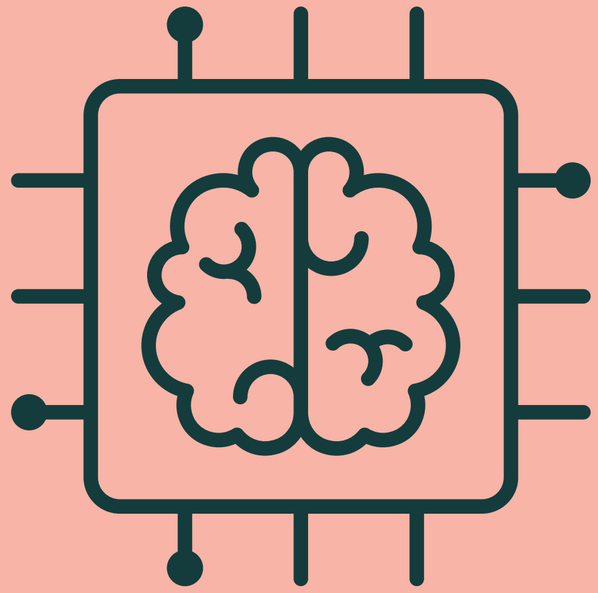








Reflections and Future Direction



Team Reflections

Task 1: Project Proposal

Our research began with the initial research proposal where we formed our groups and began work on the initial research direction. We spent time particularly researching background literature in cyber security and homomorphic encryption and chose the direction of Homomorphic encryption systems.

Feedback

During this time as a team the main obstacles with our research was that our scope was too broad. At this point we had decided on the homomorphic encryption systems but we lacked the direction in terms of application such as if it could work for Text data, Images, Videos Etc. We also spent some time to further narrow and define the specific Research problem we were trying to solve.

Team Reflections

Task 2: Project Proposal Pitch

During task 2 we further refined the research pitch and implemented the models and testing. We implemented the initial solution to our research problem and were able to tweak the models and refine the systems.

Feedback

Some key challenges we faced for this assignment were that we had some trouble finding the appropriate datasets and performing the experimental analysis. As a team.

Another challenge we found was the creation of a research paper and proposal. In particular we needed to really define the purpose of the paper and figure out the key audience in order to cut down the general and unneccesary information.

Improvements

Clear communication channels

During future research projects we will maintain clear communication channels so that the team can collaborate and work in predictable and structured ways. As we conducted our research there was often times where our efficiency would decrease because had not communicated regularly. One implementable change we could make is create specific scheduled weekly meetings.

Defining roles and responsibilities

One area we could have improved on is that we did not use a structured form of technology to define the responsibilities of each team member. There are many pieces of software such as trello boards etc that we could have used to distribute specific tasks and make them more actionable.

Collaboration and feedback

As a team we had some difficulties finding a good way to collaborate and give feedback to each other. After we reflected on this problem we needed to increase the frequency of feedback received from each other and on a regular basis look for collaboration opportunities with the group members to get ideas or help with challenges during the research.

Future Research Directions

Homomorphic encryption is a rapidly evolving technology that has many potential usages and optimisations. Our model demonstrates a highly achievable solution.

1. **Increased Efficiency:** Further developments should focus on reducing performance overheads in terms of speed and resource usage, allowing for practical implementation in practical scenarios.
2. **Standardization Efforts:** There is a need for a standardised encryption system in order to enable interability across industries and companies.
3. **3rd Party Software:** developing user-friendly tools and libraries that simplify the use of homomorphic encryption will enable widespread adoption of the technology.
4. **Scalability:** Future research should focus on improving the scalability of homomorphic encryption in cases where datasets could be significantly larger than our current test data. In cases of datasets containing greater magnitudes of data entries this could be extremely significant.

Thank you!