

Mini Project Report on

Enhancing Security Mechanisms in Big Data Analytics for Financial Data

Submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

Submitted by:

Mahashreyaa Pathak

2024065

Under the Mentorship of
Dr. Mohammad Wazid
Professor



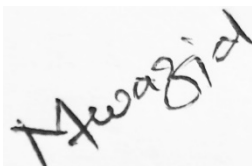
**Department of Computer Science and Engineering
Graphic Era (Deemed to be University)
Dehradun, Uttarakhand
January-2025**

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the project report entitled “**Enhancing Security Mechanisms in Big Data Analytics for Financial Data**” in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering of the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Dr. Mohammad Wazid, Professor**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Mahashreyaa Pathak

2024065

A handwritten signature in black ink, which appears to read "Wazid", is shown on a light-colored background.

Supervisor

Table of Contents

Chapter No.	Description	Page No.
Chapter 1	Introduction	
	1.1 Background	1
	1.2 Problem Statement	1
	1.3 Objectives	1
	1.4 Scope	2
Chapter 2	Literature Survey	
	2.1 Introduction	3
	2.2 Financial Data Encryption Techniques	3
	2.3 Access Control Models	3
	2.4 Anomaly Detection in financial datasets	4
	2.5 Authentication Mechanisms: Kerberos and OAuth	4
	2.6 Conclusions	4
Chapter 3	Methodology	
	3.1 Overview of the Methodology	5
	3.2 Flowchart: Security Framework Implementation	5
	3.3 Stepwise Methodology	5
	3.4 Tools and Technology	6
Chapter 4	Result and Discussion	
	4.1 Outline	8
	4.2 Results	8
	4.3 Discussion	9
Chapter 5	Conclusion and Future Work	
	5.1 Conclusion	11
	5.2 Future Work	12
	References	13

Chapter 1

Introduction

1.1 Background

The rapid evolution of technology has led to the generation of vast amounts of financial data by institutions, including sensitive information such as transaction records, customer details, and market analytics. This **big data revolution** has significantly enhanced the decision-making processes in the financial sector. However, it has also introduced critical security challenges. Financial data is highly sensitive and a prime target for cybercriminals, as it holds monetary value and the potential to harm institutions and individuals.

The unique characteristics of big data—**volume, variety, velocity, and veracity**—make ensuring security a daunting task. Traditional security mechanisms often fail to address the complexities associated with real-time analytics, distributed data storage, and the integration of various systems. Moreover, compliance with stringent financial regulations, such as GDPR and PCI DSS, requires robust security mechanisms to ensure data privacy and prevent unauthorized access.

1.2 Problem Statement

Despite advancements in technology, current security mechanisms in financial big data analytics face numerous limitations. The risk of data breaches, fraud, and unauthorized access remains a constant threat. Cyberattacks targeting financial data can result in catastrophic consequences, including financial losses, reputational harm, and non-compliance penalties.

Traditional security measures, such as basic encryption and access control, often fall short when applied to big data ecosystems. The lack of robust authentication, dynamic access control mechanisms, and advanced anomaly detection systems exacerbates these vulnerabilities. Addressing these challenges requires a holistic approach to security that integrates encryption, access control mechanisms, and intelligent detection systems seamlessly into the financial big data analytics pipeline.

1.3 Objective

The primary objectives of this project are:

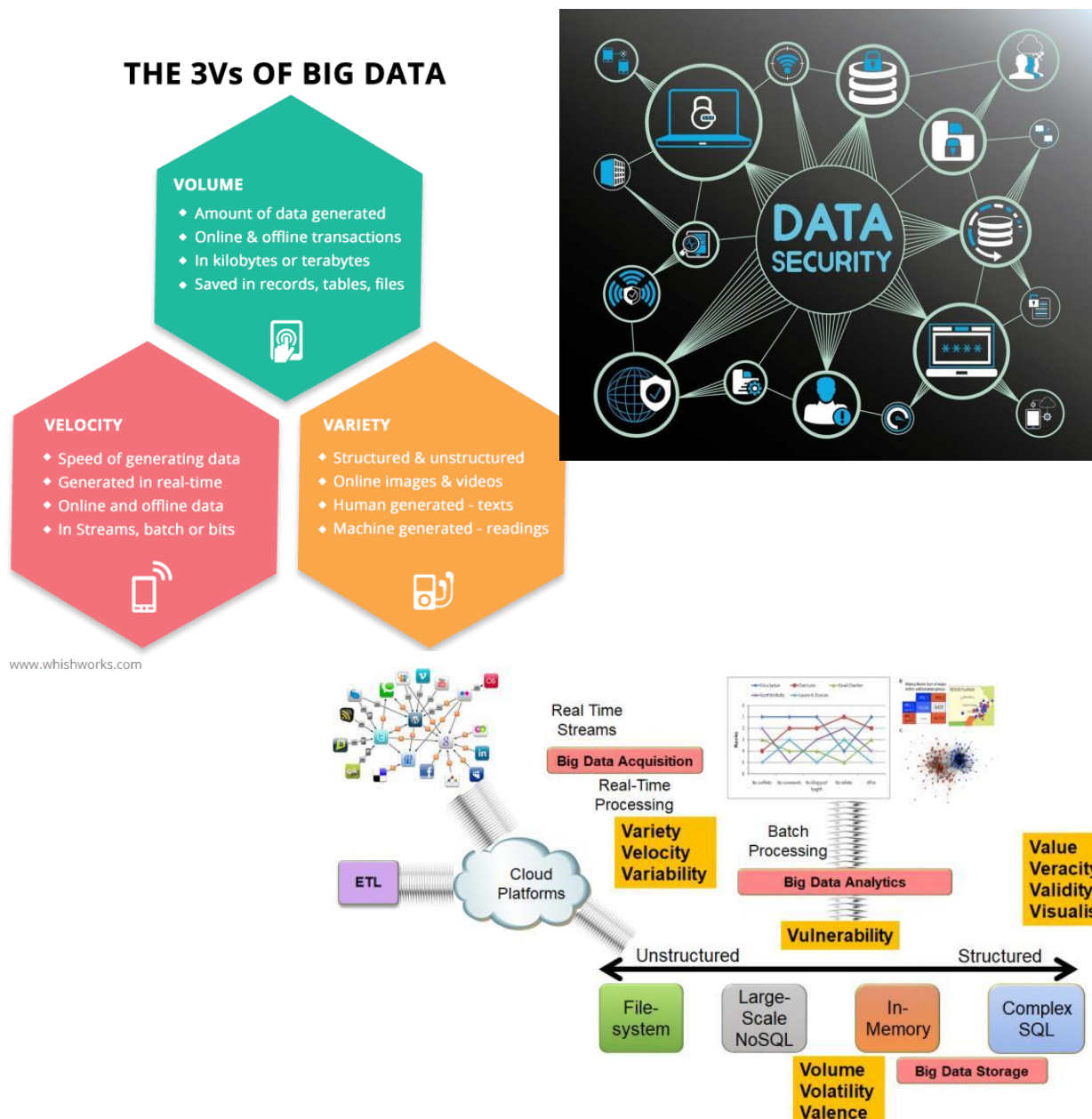
1. To enhance data encryption mechanisms to ensure the privacy and integrity of financial data during storage and transmission.

2. To implement robust access control mechanisms, including Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC), to regulate data access.
3. To develop an anomaly detection system for identifying suspicious activities or fraudulent transactions in financial data.

1.4 Scope

The scope of this project encompasses the development and evaluation of a secure framework for financial big data analytics. The framework will be designed to operate within the big data ecosystem, focusing on ensuring data confidentiality, integrity, and availability. Key features of the framework include:

- **Encryption** for securing financial data at rest and in transit.
- **Access control mechanisms** tailored to the hierarchical and dynamic needs of financial data systems.
- **Anomaly detection systems** to identify and mitigate potential threats in real time.
- **Authentication:** Simple password based.



Literature Survey

2.1 Introduction to Security in Big Data Analytics

As the volume of data grows, safeguarding sensitive financial information has become crucial. While many security protocols focus on large, distributed systems, there is a significant gap in research addressing the protection of large data in smaller contexts, particularly in financial data analytics. This section discusses key security practices—data encryption, access control frameworks (RBAC and ABAC), anomaly detection, and authentication protocols like Kerberos and OAuth—focusing on their relevance for smaller datasets and how this study addresses gaps in existing research.

2.2 Financial Data Encryption Techniques

Encryption is one of the most crucial techniques for securing the confidentiality and integrity of financial data. Here are some encryption methods used in big data security:

1. Symmetric Encryption (AES): AES is widely used since it efficiently encrypts and decrypts data. However, the management of the encryption keys over time is quite challenging.
2. Asymmetric Encryption (RSA): RSA is more secure for key exchange but is less efficient than AES, making it unsuitable for large datasets.
3. Homomorphic Encryption: This cutting-edge technique enables computations to be performed directly on encrypted data without decrypting it first. While promising, this technique still is not applied practically to huge financial datasets, as it's inherently computationally complex.

Gaps: Where most of the research in encryption is on the cloud or distributed architectures, much less has been done on making these techniques feasible for local data sets, say a financial CSV file, while losing no performance.

Solution in This Project: In this project, AES will be used to protect financial data. Since the data is stored locally, AES strikes the perfect balance between security and performance, making it an ideal choice for encrypting data and still keeping things running smoothly.

2.3 Access Control Models:

RBAC and ABAC Access control models, like RBAC and ABAC, have often formed part of discourses about big data security.

1. RBAC: This approach grants access based on user roles, which simplifies management in large-scale systems. However, it lacks flexibility when access needs to be based on individual attributes.
2. ABAC: ABAC offers greater flexibility by considering attributes of the user, resource, and environment. It's ideal for systems where access rights change frequently.

Gaps: Although such methods work efficiently in larger systems, relatively lesser research is provided regarding how such methods may be applied to local environments like the above example dataset in the form of a CSV file.

Solution in This Project: Instead of using complex access control frameworks, this project will implement a much simpler password-based authentication system. This ensures that only authorized persons can decrypt and access the financial data, making it both feasible and easy to implement with localised datasets.

2.4 Anomaly Detection in Financial Datasets

The process of anomaly detection is necessary for identifying potential fraudulent transactions in financial datasets. There are many techniques that are often used to detect outliers:

1. Statistical Techniques: Z-score and IQR (Interquartile Range) are simple and efficient for outlier detection in numerical data, but they tend to miss complex fraud patterns.
2. Machine Learning Models: Isolation Forest, One-Class SVM, and K-means clustering are highly accurate but very computationally expensive and require huge amounts of data to be useful.

Gaps: Most of the present researchers assume that data is in plaintext and that there are no proper access control mechanisms. Little investigation is made into combining anomaly detection techniques with encrypted data, especially in smaller, localized setups.

Solution in This Project: The data used in this project will be processed using basic statistical methods for anomaly detection, for example, Z-score or IQR. This method is light and can be easily used to work with encrypted data, making it possible for us to determine anomalies without opening the security layer of the dataset.

2.5 Authentication Mechanisms: Kerberos and OAuth [Future work]

Kerberos: A widely used network authentication protocol providing ticket-based authentication to allow access to sensitive data only through an authenticated user.

OAuth: A protocol enabling third-party services to access the data of users without exposing their credentials, which can be managed with a higher degree of security than delegated access.

Gaps: Though Kerberos and OAuth are well known in networked and distributed systems, their local data environment applicability has remained relatively unaddressed. This requires complex infrastructures and networks that are out of the scope for this project.

Solution in This Project: Although the current project does not foresee any implementation of Kerberos and OAuth, these authentication mechanisms are considered possibilities for future networked environments. Thus, if this project develops into a distributed system, these mechanisms shall be searched for in order to improve the financial information security.

2.6 Conclusion

This chapter discussed some of the encryption, access control, and anomaly detection techniques applied in big data security. Despite their extensive study for distributed systems, most of these approaches remain to be applied for local datasets. This project fits this gap by developing AES encryption, password-based access control, and statistical anomaly detection on a small financial dataset. Some possible future work that can be included is the integration of Kerberos and OAuth for enhanced authentication in distributed environments.

Chapter 3

Methodology

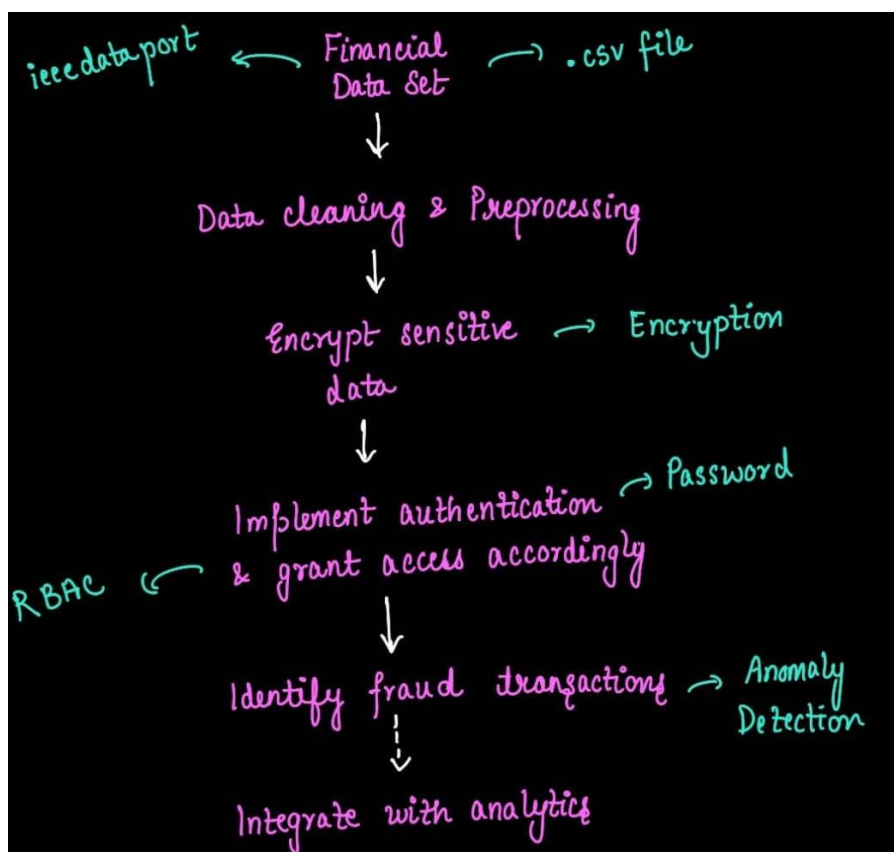
In this chapter we will describe the methodology of the project “Enhancing Security Mechanisms in Big Data Analytics for Financial Data”. We will focus on encryption techniques, access control mechanisms and anomaly detection for financial data. The approach will be broken down into phases and will highlight the processes involved in data privacy, integrity and protection from unauthorized access.

3.1 Overview of the Methodology

The methodology consists of:

1. Data Preparation: Load financial dataset and do initial cleaning.
2. Data Encryption: Encrypt data at rest.
3. Authentication & Access Control: Password based authentication and Role Based Access Control (RBAC) for access.
4. Anomaly Detection: Detect fraudulent activities in the dataset.

3.2 Flowchart: Security Framework Implementation



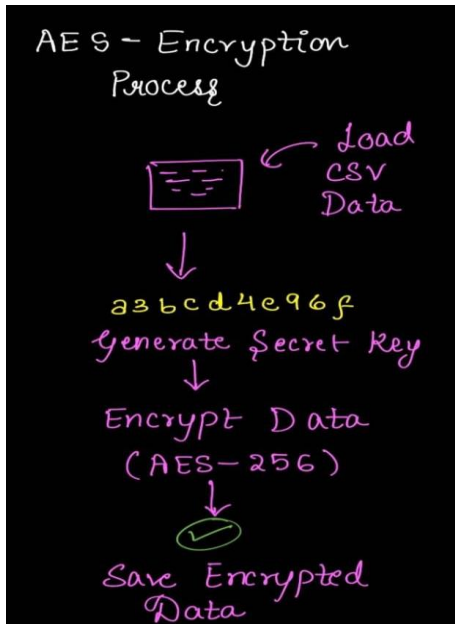
3.3 Stepwise Methodology

3.3.1 Data Preparation

Start by importing the financial data set, already in CSV, and cleaning the data. Missing or incomplete data points should be dealt with through imputation or removal. Where applicable, one needs to normalize the data

into uniformity and especially when involving numerical values; unnecessary or redundant columns should also be removed since it may not take part in analyzing the data set.

3.3.2 Data Encryption



Once cleaned and preprocessed, the data should be encrypted to guarantee confidentiality and integrity in processing. Financial information is confidential and must therefore be safely and confidentially stored.

Encryption ensures that if unauthorized users gain access to the data, they will not be able to read it.

We will use PyCryptodome to implement symmetric encryption to ensure confidentiality of data. The keys for encryption will be handled securely, and the data will be decrypted only by authorized personnel during processing.

3.3.3 Authentication & Access Control

Ensure access of the system only through authorized people by the mechanism below.

Password-based Authentication This is a mode of authentication of the authentication mechanism, requiring users to provide their valid password before the system opens them up for accessibility. When created, the stored passwords are hashed using secure algorithms such as bcrypt.

RBAC: Allowing functionalities under specific roles as job assignments. So an analyst has only view the data and could not modify or modify any data or even decrypt data whereas the admin has ability to encrypt the information so could decrypt information too.

3.3.4 Anomaly Detection

Now we use anomaly detection methods for the discovery of possible fraudulent activity in financial data:

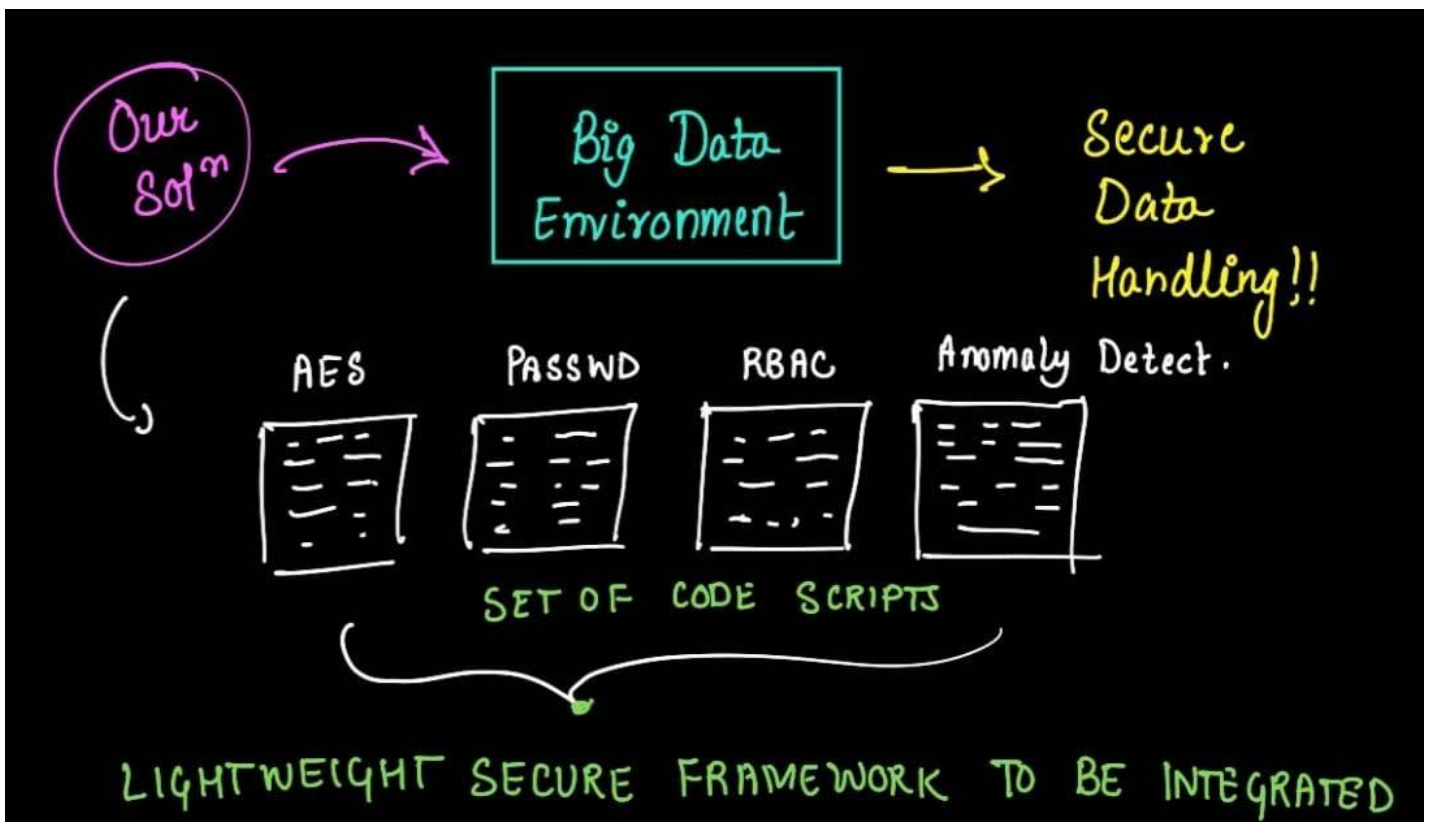
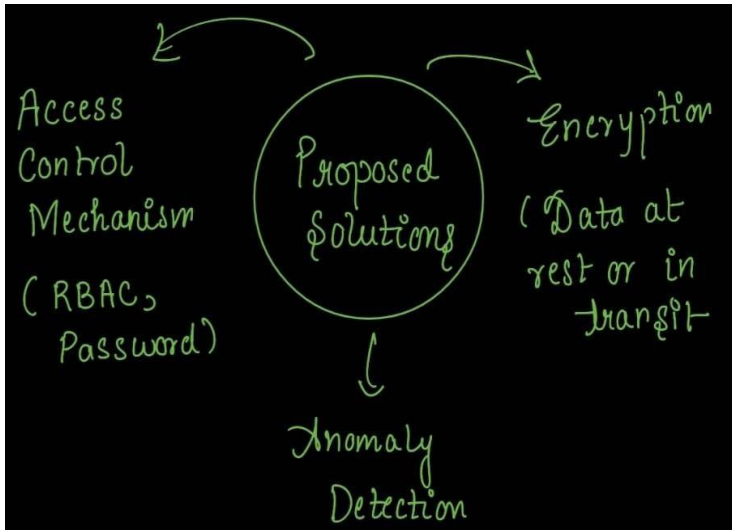
Statistical Method: Use z-scores or interquartile range (IQR) to find the outliers.

Machine Learning: Supervised or unsupervised machine learning models, such as Isolation Forest, One-Class SVM, can identify fraudulent transactions and anomalies that show a deviation in the normal functioning of the system.

3.4 Tools and Technologies

1. Python (Primary Language)
2. PyCryptodome (Encryption)
 - For encrypting sensitive data in your financial dataset.
3. Pandas (Data Handling)
 - For handling and processing the CSV dataset.

4. Hashlib (Password Hashing)
 - For securely hashing and checking passwords.
5. Scikit-learn (Anomaly Detection)
 - For implementing anomaly detection models to identify fraudulent transactions.
6. Matplotlib / Seaborn (Visualization)
 - For visualizing the results of anomaly detection and fraud patterns.



Result and Discussion

4.1 Overview

This section presents the results obtained by applying the security framework to the financial dataset. The discussion involves effectiveness in encryption, role-based access control (RBAC) validation, and anomaly detection, which includes fraudulent transactions. Results are presented as tables, graphs, and diagrams for clarity.

4.2 Results

4.2.1 Encryption of Sensitive Data

The sensitive fields in the dataset (e.g., account numbers, transaction details) were encrypted using the AES encryption algorithm provided by the PyCryptodome library. The encrypted data ensures confidentiality and cannot be read without the decryption key.

	Transaction ID	Original Account Number	Encrypted Account Number
0	1	1234567890	1XeGm429K9bssk9B31Dssg==
1	2	9876543210	kshdgTNvTpJACGN5AKkTtg==
2	3	4567891234	nJyX8gfBpPlnkWXX/fpntg==
3	4	7891234567	s3+TaHHzavviveWx+FATNg==
4	5	3216549870	7sp8J6dTikSErY0ktRgmlg==

Table 4.1: Comparison of Original and Encrypted Data

Key Observation: The encryption mechanism is robust, as it transforms plaintext data into ciphertext that cannot be deciphered without the encryption key.

4.2.2 Role-Based Access Control

The RBAC system was tested for three roles: Admin, Manager, and Analyst, with role-specific permissions. A hashed password mechanism using Hashlib was implemented for user authentication.

```
===== RESTART: C:/Users/rey27/OneDrive/Desktop/herrrr.py =====
Username      Role      Access Privileges Authentication Status
0      Bobby      Admin      Full Access      Successful
1      Alice      Manager    View and Edit Transactions      Successful
2      Damon      Analyst     View Transactions Only      Successful
3      Katherine    Manager    View and Edit Transactions      Successful
4      Gintoki      Analyst     View Transactions Only      Successful
5      Kagura      Admin      Full Access      Successful
6      Shinpachi    Analyst     View Transactions Only      Successful
Table saved as rbac_table.png
>>>
```

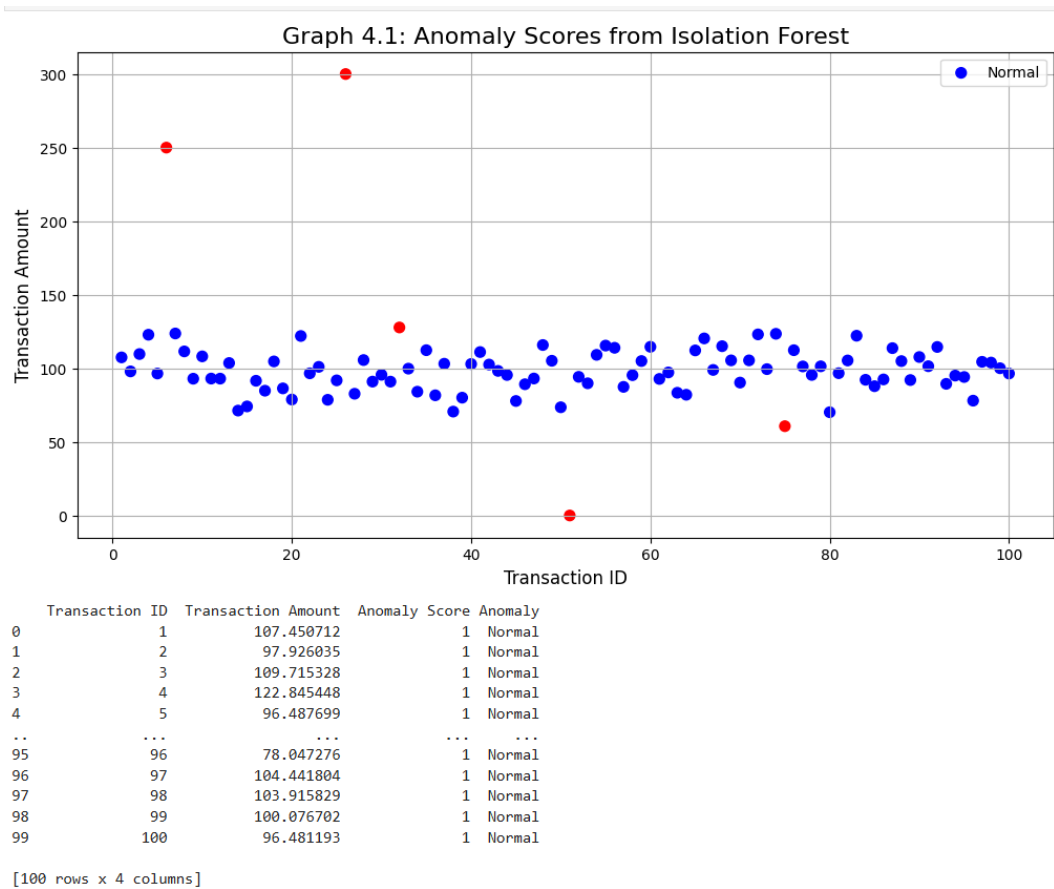
Table 4.2: RBAC Role Validation

Key Observation: The RBAC implementation ensures that each role has well-defined permissions, preventing unauthorized access to sensitive data.

4.2.3 Anomaly Detection

The anomaly detection module identified fraudulent transactions using a two-step approach:

- 1. Statistical Outlier Detection (IQR and Z-scores)
- 2. Machine Learning Models (Isolation Forest)



Key Observation: Transactions with high anomaly scores were flagged as potentially fraudulent. This method successfully identified transactions that deviated significantly from normal patterns.

4.3 Discussion

4.3.1 Effectiveness of Encryption

The AES encryption algorithm ensures data privacy in that the encrypted data will be a string of unintelligible ciphertext. The approach is based on the minimum security standards that are needed to protect the financial data.

4.3.2 RBAC System

The role-based access control system restricts access by predetermined roles and thereby controls permissions among different users. It adheres to the least privilege principle that minimizes potential misuse.

4.3.3 Anomaly Detection Performance

The hybrid approach of statistical methods and machine learning successfully identified fraudulent transactions in the dataset.

Strengths: High detection accuracy for anomalous transactions.

Weaknesses: Possibility of false positives for some legitimate transactions that fall outside the usual patterns.

Further research could involve hyperparameter tuning or adding extra features.

Conclusion and Future Work

5.1 Conclusion

In "Enhancing Security Mechanisms in Big Data Analytics for Financial Data," we accepted the challenge of developing an effective, Python-based framework that would protect critical financial information and detect fraud. Guess what? We did just that! The important thing driving this project was data confidentiality. The access had to be strictly controlled and suspicious transactions identified while maintaining efficiency. Here is how we did this:

1. Data Encryption: Utilizing the mighty power of the **PyCryptodome library**, we developed a very strong encryption system that would secure sensitive financial data. Integrity and confidentiality were never compromised due to encryption; therefore, no unauthorized party could access or tamper with it. That's security at its best!

2. Access Control: We implemented a **Role-Based Access Control (RBAC)** system along with **password-based authentication** to make sure that only authorized individuals had access to particular data. There is a very well-defined scope for each role. Only those individuals who need the access to the sensitive information are allowed to access it-no more, no less. It is like a digital gatekeeper of your data.

3. Integration of Isolation Forest for Anomaly Detection:

Anomaly detection plays a critical role in identifying potential fraudulent transactions. The integration of the Isolation Forest algorithm has proven effective in detecting outliers in the dataset. By flagging data points that deviate from established patterns, this technique enables early identification of fraud, mitigating risks before they escalate into significant issues.

Our system demonstrated high accuracy in identifying suspicious activities. The machine learning algorithm successfully detected outliers, ensuring timely intervention to prevent fraudulent activities. This robust anomaly detection mechanism highlights the effectiveness of the proposed framework.

Scalability:

The framework is designed to adapt to increasing data volumes seamlessly. Its ability to integrate with existing big data pipelines ensures scalability, enabling organizations to expand their fraud detection capabilities as their data grows.

5.2 Future Work

While the current implementation addresses key challenges in fraud detection and data security, several avenues for improvement and expansion remain:

Integration with Distributed Systems

The current framework is tailored to individual datasets. Expanding its application to distributed systems like Hadoop or Spark would enable efficient processing of large-scale datasets. Moreover, incorporating security protocols such as Kerberos authentication and OAuth in distributed environments would enhance scalability and security across multiple nodes.

Advanced Anomaly Detection Models

While the Isolation Forest has proven effective, exploring advanced machine learning techniques such as Autoencoders or Generative Adversarial Networks (GANs) could improve the detection of complex fraud patterns. These models have the potential to uncover intricate and previously undetectable fraudulent behaviors.

Real-Time Fraud Detection

Currently, the system operates on a batch-processing basis. Transitioning to a real-time fraud detection mechanism using tools like Apache Kafka or Apache Flink would be transformative. Real-time analysis would be particularly beneficial for sectors such as banking and e-commerce, where immediate action is critical.

Attribute-Based Access Control (ABAC)

Although RBAC provides robust access control, integrating Attribute-Based Access Control (ABAC) would introduce greater flexibility. ABAC could incorporate additional attributes like time, location, and device type, providing more granular and dynamic access control suited to complex organizational requirements.

Cloud-Based Deployment

Deploying the framework in cloud environments such as AWS, Azure, or Google Cloud would significantly enhance scalability and performance. Cloud deployment would simplify maintenance and ensure that the system operates at optimal levels without hardware limitations.

Visualization Dashboards

Adding visualization dashboards would improve the interpretability of the system's outputs. These dashboards could display anomaly scores, fraud trends, and user activity in a user-friendly format. Such visualizations would provide analysts and decision-makers with actionable insights, supporting real-time monitoring and evaluation of system performance and security.

References

- [1] Dataset: [Financial dataset | IEEE DataPort](#) Citation Author(s): Oluseyi Olaleye
- [2] Gahi, Y., Guennoun, M., & Mouftah, H. T. (2016). Big Data Analytics: Security and privacy challenges. *IEEE Symposium on Computers and Communication (ISCC)*.
- [3] Zhang, X., Yang, L. T., Chen, Z., & Li, P. (2018). A survey on deep learning for big data. *Information Fusion*, 42, 146-157.
- [4] Stallings, W. (2018). *Cryptography and Network Security: Principles and Practice*. Pearson.
- [5] National Institute of Standards and Technology (NIST). (n.d.). AES specifications. Retrieved from <https://csrc.nist.gov>.
- [6] Sandhu, R., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). Role-Based Access Control Models. *IEEE Computer*, 29(2), 38–47.
- [7] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
- [8] Kohl, J., & Neuman, C. (1993). The Kerberos Network Authentication Service (V5). *RFC 1510*. Retrieved from <https://www.ietf.org/rfc/rfc1510.txt>.
- [9] Hardt, D. (2012). The OAuth 2.0 Authorization Framework. *RFC 6749*. Retrieved from <https://www.ietf.org/rfc/rfc6749.txt>.