**Real-Time AML Suspicious Pattern Clustering**

**A PROJECT REPORT**

***Submitted by***

**MUTHULAKSHMI M 2116231801114**

**RAMYA A 2116231801135**

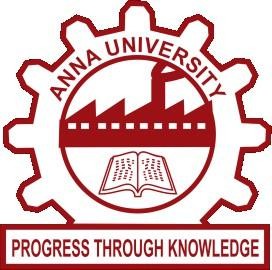
**MEENAKSHI G 2116231801099**

***in partial fulfillment for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

***in***

**ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

****

**RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS), CHENNAI – 602 105**

**OCTOBER 2025**

**BONAFIDE CERTIFICATE**

Certified that this Report titled “**Real-Time AML Suspicious Pattern Clustering”**

is the Bonafide work of “**MUTHULAKSHMI M (2116231801114) RAMYA A (2116231801135) MEENAKSHI G (2116231801099)”** who carried out the work

under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**Mr. Subramanian K ,**

**Professor,**

Department of Artificial Intelligence & Data Science,

Rajalakshmi Engineering College

Thandalam – 602 105.

Submitted to Project Viva-Voce Examination held on

**Internal Examiner External Examiner**

# ACKNOWLEDGEMENT

Initially I thank the Almighty for being with us through every walk of my life and showering his blessings through the endeavor to put forth this report.

My sincere thanks to our Chairman **Mr. S. MEGANATHAN, M.E., F.I.E.,** and our Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN, M.E., Ph.D.,** for providing me with the requisite infrastructure and sincere endeavoring educating me in their premier institution.

My sincere thanks to **Dr. S.N. MURUGESAN M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time.

I express my sincere thanks to **Dr. J M Gnanasekar M.E., Ph.D.,** Head of the Department of Artificial Intelligence and Data Science for his guidance and encouragement throughout the project work. I convey my sincere and deepest gratitude to our internal guide, **Mr. Subramanian K ,** Professor, Department of Artificial Intelligence and Data Science, Rajalakshmi Engineering College for his valuable guidance throughout the course of the project.

Finally, I express my gratitude to my parents and classmates for their moral support and valuable suggestions during the course of the project.

### MUTHULAKSHMI M RAMYA A MEENAKSHI G

### (2116231801114) (2116231801135) (2116231801099)

### 

### 

### ABSTRACT

Electricity theft remains a pervasive global challenge that causes substantial financial losses, destabilizes power grids, and affects the overall quality and reliability of electricity distribution systems. Traditional detection techniques, such as manual meter inspections, audits, or consumer complaint investigations, are time-consuming, error-prone, and lack the scalability required for modern, data-intensive smart grid infrastructures. With the rapid deployment of smart meters, vast volumes of energy consumption data are continuously generated, creating an unprecedented opportunity for intelligent, automated theft detection through Big Data analytics and machine learning. This project introduces a comprehensive Big Data–driven **Energy Theft Anomaly Detection System**, implemented on the **Google Cloud Platform (GCP)** using technologies such as **Hadoop**, **PySpark** on **Dataproc**, **BigQuery**, and **Looker Studio**. The system efficiently ingests raw smart meter readings into Google Cloud Storage, preprocesses them using distributed PySpark computations, extracts key consumption features, and applies statistical and machine learning–based anomaly detection techniques to identify suspicious consumption behaviors that may indicate theft. Processed and scored data are then stored in BigQuery for scalable, high-speed querying and subsequently visualized through dynamic and interactive dashboards in Looker Studio, enabling clear interpretation and actionable insights. By unifying data ingestion, preprocessing, analytical modeling, and visualization under a cloud-based architecture, the system achieves high scalability, fault tolerance, and real-time responsiveness. Ultimately, this project empowers energy distribution companies to transition from reactive to proactive monitoring, reduce non-technical losses, enhance grid security, and ensure equitable energy distribution through intelligent, data-driven decision-making powered by modern Big Data technologies.

**TABLE OF CONTENTS**

**CHAPTER NO NAME PAGE NO**

### Background 7

### Motivation 7

### Objectives 8

### Problem Statement 9

### Scope of the Project 10

### 2.1 Traditional Methods 11

### 2.2 Statistical and Rule-Based 11

### Techniques

### 2.3 Machine Learning 11

### 2.4 Big Data and Cloud-Based 11

### Approach

### 2.5 Anomaly Detection 12

### 2.6 Summary of Research Gaps 12

### 2.7 Contribution of the present 14 work

### 3.1 System Overview 15 3.2 System Architecture 15

### 3.3 System Requirment

### 3.3.1 Hardware 17

### 3.3.2 Software 17

### 

### 3.4 System Modules 18

### 3.5 Data Flow Diagram 19

### 3.6 Summary 20

### 4.1 Data Collection Module 21 4.2 Data Preprocessing Module 21

### 4.3 Hive Query & Hive Table 23

### Creation

### 4.4 Visualization Module 24

### 4.5 Dashboard Module 24

### 5.0 IMPLEMENTATION 25

### 6.0 RESULTS 26

### 7.0 CONCLUSION 29

### 8.0 FUTURE ENHANCEMENTS 30

### REFERENCES 31

### LIST OF FIGURES

**FIGURE NO NAME PAGE NO**

1. **Big Data–Based Energy 13  
   Theft Detection Architecture   
   on Google Cloud Platform**
2. **Crime Hotspots Analysis 22  
   Dashboard**

### CHAPTER 1

### INTRODUCTION

#### Background

Money laundering has become one of the most critical challenges in the global financial ecosystem. It involves disguising the origins of illegally obtained funds through complex financial transactions, making the money appear legitimate. According to the United Nations Office on Drugs and Crime (UNODC), the estimated amount of money laundered globally each year is between **2% and 5% of global GDP**, equivalent to **$800 billion to $2 trillion USD**. Financial institutions, therefore, face increasing pressure to detect and prevent such illicit activities to maintain transparency and comply with international regulations such as the **Financial Action Task Force (FATF)** guidelines and **Anti-Money Laundering (AML)** directives.

Traditional AML detection systems primarily rely on **rule-based mechanisms** and **manual reviews**, where predefined thresholds or conditions trigger alerts. For example, transactions exceeding a specific amount or frequent cross-border transfers are flagged for investigation. However, these systems often result in **high false-positive rates**, require extensive human intervention, and struggle to adapt to evolving laundering strategies that exploit technological loopholes and diverse transaction channels such as online banking and fintech platforms.

With the rapid digitization of financial services, the volume, variety, and velocity of transaction data have grown exponentially. Each second, thousands of digital transactions occur across multiple countries, channels, and currencies, generating massive, heterogeneous datasets. To analyze such data efficiently and detect hidden suspicious patterns, **Big Data technologies** and **Machine Learning (ML)** models have emerged as powerful alternatives.

By integrating **Apache Spark**, **Kafka**, and distributed cloud platforms, financial institutions can now process real-time transactional streams and identify anomalies that deviate from normal behavior. Machine learning techniques, such as **DBSCAN clustering** and **Isolation Forest**, enable unsupervised detection of suspicious activity without prior labeling, revealing potential money laundering networks and behavioral outliers automatically.

This data-driven approach transforms traditional AML monitoring into a **real-time, intelligent surveillance system**, capable of learning from dynamic transaction behavior, reducing false positives, and providing timely alerts to compliance officers. Through the combination of **Big Data analytics**, **streaming pipelines**, and **AI-driven anomaly detection**, banks can significantly enhance their ability to combat financial crime while ensuring compliance with global AML standards.

#### Motivation

Money laundering remains one of the most critical challenges faced by financial institutions across the world. As digital banking, online payments, and international transactions continue to grow, financial crimes have also become more sophisticated and harder to detect. Criminals often disguise illegal funds by passing them through multiple accounts and channels, making it difficult to trace their origin. This not only threatens the integrity of the global financial system but also impacts economic stability and public trust in banking institutions.

Existing AML systems mostly rely on predefined rules and manual inspection processes, which are time-consuming and limited in their ability to adapt to new or unknown fraud patterns. These rule-based models often lead to a high number of false alerts, increasing the workload for investigators while still missing complex money-laundering activities that don’t follow known behaviors. Moreover, traditional systems lack the computational power to handle the enormous volume and speed of modern financial data, especially when real-time analysis is needed.

This project is motivated by the need for a more **intelligent, scalable, and adaptive solution**. By utilizing **Big Data analytics** and **machine learning-based clustering techniques** such as DBSCAN, the proposed system can analyze large transactional datasets in real time and automatically identify abnormal patterns that may indicate potential laundering activities. The integration of Apache Spark ensures that the system can process millions of transactions efficiently, providing faster insights and reducing manual intervention. Ultimately, this approach aims to strengthen financial security, improve fraud detection accuracy, and contribute to a more transparent and trustworthy banking ecosystem.

#### Objectives

#### The main objective of this project is to develop a Big Data-driven, real-time system for detecting suspicious financial transaction patterns related to money laundering. The project aims to analyze large-scale banking data efficiently using distributed frameworks like Apache Spark, and apply unsupervised machine learning techniques such as DBSCAN clustering to automatically group and identify abnormal behaviors. By focusing on key features like transaction amount, frequency, location, and cross-border activity, the system seeks to uncover hidden relationships and unusual transaction clusters that may indicate potential fraud. Additionally, the project aims to reduce false positives, enhance detection accuracy, and provide scalable, automated, and intelligent support for financial institutions to strengthen their AML operations and regulatory compliance.

#### Problem Statement

With the rapid rise in digital banking, mobile payments, and global financial transactions, money laundering activities have become increasingly complex and harder to detect. Traditional Anti-Money Laundering (AML) systems rely heavily on static, rule-based methods that flag transactions based on fixed thresholds or known fraud indicators. While effective for known patterns, these systems fail to recognize **new, evolving, or hidden suspicious behaviour** that do not fit predefined rules.

Additionally, the continuous flow of massive, real-time financial data from various channels — such as online banking, cross-border transfers, and mobile applications — poses a serious challenge for traditional systems that lack scalability and adaptability. These systems struggle to process large datasets efficiently, leading to **delayed analysis, false alerts, and missed detections**.

Therefore, there is a strong need for an **intelligent, data-driven, and scalable AML detection system** capable of analysing real-time transactional data and automatically identifying suspicious patterns without manual intervention. By leveraging **Big Data analytics** and **unsupervised machine learning techniques** like **DBSCAN clustering**, such a system can effectively detect anomalies, reduce false positives, and support real-time financial decision-making.

**1.5 Scope of the Project**

The scope of this project lies in developing an intelligent, data-driven system that can detect suspicious financial activities in real time using Big Data analytics and clustering techniques. The system focuses on analyzing large-scale transactional data from multiple banking sources, including online payments, ATM withdrawals, mobile banking, and international fund transfers. By leveraging distributed computing frameworks such as **Apache Spark**, the system can process millions of records efficiently, allowing for continuous monitoring and detection of unusual transaction behaviors.

Unlike traditional rule-based AML systems, this project emphasizes the use of **unsupervised machine learning algorithms**, particularly **DBSCAN**, to automatically identify clusters of similar transactions and isolate anomalies that deviate from normal financial activity. The model is capable of adapting to dynamic patterns without relying on predefined thresholds, making it more flexible and effective in recognizing new and evolving money-laundering techniques. This intelligent clustering approach provides deeper insights into transaction behaviors, helping financial institutions strengthen their internal security systems and minimize false alerts.

Furthermore, the project’s scope extends toward scalability, integration, and future enhancement. It can be implemented within modern banking infrastructures and scaled up using cloud-based Big Data platforms such as **AWS**, **GCP Dataproc**, or **Azure Synapse**. In the future, additional components such as visualization dashboards, real-time alert systems, and predictive modeling can be integrated to provide end-to-end AML intelligence. Overall, this project lays the foundation for building a **real-time, adaptive, and automated AML framework** capable of ensuring financial transparency, reducing operational risk, and improving global compliance standards.

**CHAPTER 2**

**LITERATURE SURVEY**

The detection of electricity theft has been an active area of research due to the significant financial and operational losses it causes in power distribution systems. Over the years, various techniques have been developed — from statistical analysis to machine learning and, more recently, Big Data–driven anomaly detection systems. This chapter summarizes the most relevant studies and approaches in this field, highlighting their methodologies, limitations, and the technological evolution leading to the present work.

#### Traditional Methods

Traditional Anti-Money Laundering (AML) systems primarily rely on **rule-based detection** and **threshold-based monitoring** to identify suspicious transactions. In this approach, financial institutions set fixed rules — such as flagging transactions exceeding a certain amount, detecting frequent transfers within a short period, or monitoring transactions involving high-risk countries. When a transaction breaks any of these predefined conditions, it is flagged for further review. Some systems also use basic statistical analysis or supervised machine learning models trained on labeled fraud data. While these methods can detect known and repetitive patterns, they struggle to recognize **complex or emerging money-laundering behaviors** that evolve over time.

However, these traditional approaches come with significant **drawbacks**. Since rule-based systems depend heavily on prior knowledge and static conditions, they cannot adapt automatically to new or unseen transaction patterns. This results in a large number of **false positives**, where legitimate transactions are flagged as suspicious, leading to wasted time and manual investigation efforts. Additionally, supervised models require accurately labeled datasets, which are often difficult to obtain due to privacy concerns and limited availability of verified laundering cases. As a result, their ability to generalize and detect subtle, hidden anomalies is limited.

Another major drawback is the **lack of scalability and real-time processing capability**. Traditional AML systems are often built on centralized databases that cannot efficiently manage the high volume and velocity of data generated by digital and cross-border banking. They process data in batches rather than streams, leading to delays in detection and response. In today’s era of instant transactions, this delay can allow illicit transfers to go unnoticed. These limitations highlight the urgent need for a **Big Data-enabled, adaptive, and intelligent AML detection system** that can process massive data in real time, learn from evolving trends, and accurately cluster suspicious behaviors with minimal human intervention.

#### Statistical and Rule-Based Techniques

In traditional Anti-Money Laundering (AML) systems, most banks use **rule-based and statistical methods** to detect suspicious transactions. In rule-based systems, certain fixed rules are created — for example, if a transaction amount is higher than a set limit, or if a customer makes too many transfers in a short time, it is marked as suspicious. Statistical methods are also used to find unusual patterns by comparing transactions with normal customer behavior. These techniques are simple, easy to understand, and useful for identifying known or common fraud patterns.

However, these methods have several **limitations**. Since the rules are fixed, they cannot easily detect new or changing money-laundering techniques. Criminals often find ways to avoid these rules by slightly changing their behavior. Statistical methods also fail when the data is too large or complex. Both systems can generate many false alerts, where genuine transactions are flagged as suspicious, creating extra work for investigators. Therefore, there is a strong need for **advanced, intelligent, and data-driven systems** that can automatically learn from data, handle large volumes, and detect hidden suspicious patterns more accurately.

#### Machine Learning Approaches

Machine Learning (ML) approaches have become an important part of modern Anti-Money Laundering (AML) systems. Unlike rule-based methods, ML models can automatically learn from past transaction data and identify complex or hidden patterns that may indicate suspicious behavior. These models can handle large amounts of data and can adapt to changes in transaction trends over time. Supervised learning techniques such as **Logistic Regression**, **Random Forest**, or **Support Vector Machines (SVM)** are often used when labeled data (fraud or non-fraud) is available. They help predict whether a new transaction is suspicious based on patterns learned from historical examples.

When labeled data is not available, **unsupervised learning** techniques like **K-Means**, **DBSCAN**, or **Isolation Forest** are used to find unusual transaction patterns or anomalies. These models can group similar transactions together and highlight those that do not fit normal behavior. Machine learning-based systems are faster, more flexible, and more accurate compared to traditional methods. They can continuously improve with new data, reduce false alerts, and provide real-time insights for financial institutions to take preventive action against money laundering activities.

#### Big Data and Cloud-Based Approaches

In recent years, big data and cloud computing have changed how organizations handle and process large amounts of data. Big data tools like Hadoop and Spark help in managing and analyzing massive datasets efficiently. They allow systems to detect patterns and anomalies in real time, which is very useful in safety monitoring systems like nuclear plant risk detection. Cloud platforms such as AWS, Google Cloud, and Azure provide flexible storage and computing power, making it easier to scale projects without heavy investment in physical infrastructure.

Cloud-based systems also enable real-time data access from multiple sources, supporting continuous monitoring and decision-making. They help in integrating IoT devices, AI models, and databases under one platform, allowing teams to collaborate and deploy solutions faster. Overall, the combination of big data and cloud computing ensures faster processing, better accuracy, and high reliability for critical applications like nuclear plant safety, where quick and accurate insights can prevent potential disasters.

#### Anomaly Detection Techniques

Anomaly detection techniques are used to identify unusual patterns or data points that differ from normal behavior. In clustering-based approaches, data is grouped into clusters based on similarity, and any data points that don’t fit well into these clusters are marked as anomalies. These methods are especially useful in complex systems like nuclear plant monitoring, where identifying abnormal sensor readings early can help prevent serious failures.

Some popular clustering techniques for anomaly detection include **K-Means**, **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)**, and **Hierarchical Clustering**. K-Means helps by forming compact groups of similar data, while DBSCAN is effective at detecting outliers that do not belong to any dense region. These algorithms can automatically separate normal and abnormal behaviors without needing labeled data, which makes them suitable for real-world applications where anomalies are rare or unknown. Overall, clustering-based anomaly detection provides a simple yet powerful way to detect risks and ensure system safety.

#### Summary of Research Gaps

Despite significant progress in energy theft detection and anomaly identification, several research gaps still exist. Many existing systems depend on traditional statistical or rule-based techniques, which struggle to adapt to dynamic and large-scale data generated by modern smart grids. These methods often fail to detect complex or evolving theft patterns that do not follow predefined rules.

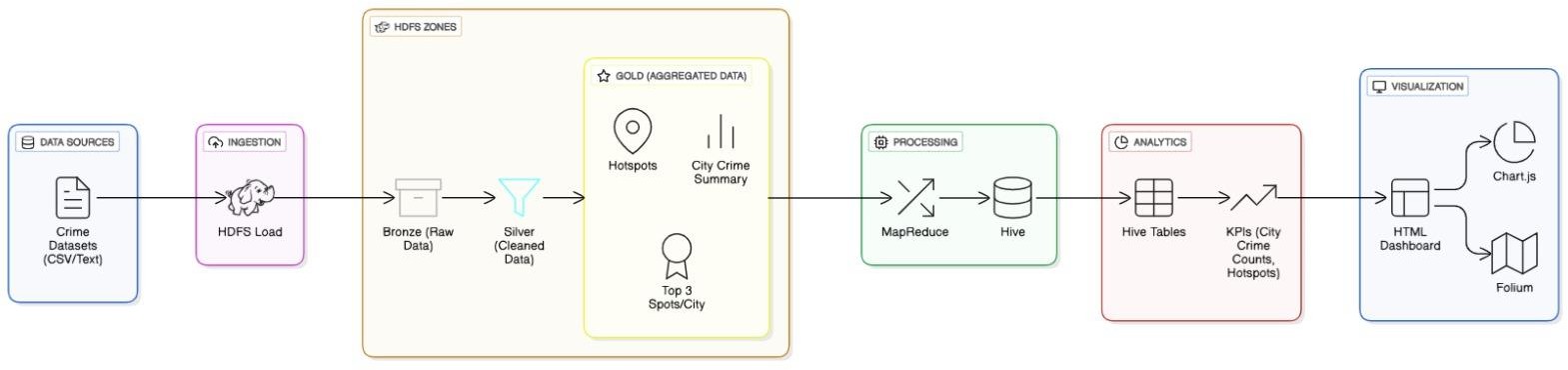
Machine learning and deep learning approaches have improved accuracy, but they still require large amounts of labeled data, which is difficult to obtain in real-world energy systems. Moreover, most studies focus on small datasets or simulations, limiting their scalability and real-time applicability. The integration of **Big Data platforms** like Hadoop and **cloud-based solutions** such as Google Cloud or AWS remains underexplored in practical implementations.

Therefore, there is a clear need for a more **scalable, intelligent, and automated system** that can handle massive energy consumption data, adapt to new theft techniques, and provide real-time detection using clustering and anomaly-based methods. Bridging this gap will help create a robust, data-driven framework for efficient and reliable energy theft detection.

#### Contribution of the Present Work

The present work focuses on developing a **Big Data–driven energy theft detection system** that leverages **machine learning and clustering-based anomaly detection techniques** to identify suspicious consumption patterns. Unlike traditional rule-based systems, this model automatically learns and adapts to new forms of irregular behavior in energy usage. The project utilizes **Apache Spark** for distributed data processing and **Google Cloud Platform (GCP)** for scalable storage and computation, ensuring efficient handling of massive datasets. By applying clustering methods such as **DBSCAN**, the system detects outliers without requiring labeled data, making it suitable for real-world scenarios.

Additionally, this work integrates **real-time data streams from smart meters**, enabling continuous monitoring of energy consumption. It contributes to the field by demonstrating how **Big Data analytics and cloud computing** can enhance detection accuracy, reduce manual efforts, and improve operational efficiency for utility companies. The project also provides a **modular framework** that can be extended for fraud detection, predictive maintenance, and other smart grid applications.

****

**Figure 1: Big Data–Based Real-Time AML Suspicious Pattern Clustering**

***Figure 1*** reflects the data flow and technologies (data ingestion, processing with PySpark/Hadoop, analytics using BigQuery, and visualization via Looker Studio)

**CHAPTER 3**

**SYSTEM ANALYSIS AND DESIGN**

This chapter explains the overall architecture, system requirements, and module design of the Energy Theft Anomaly Detection System. The goal of the system is to analyze smart meter consumption data on a large scale using Big Data tools and detect abnormal usage patterns that indicate possible energy theft.

The system leverages **Google Cloud Platform (GCP)** for storage, processing, and visualization of data using a fully integrated and scalable architecture.

#### System Overview

The proposed system is designed to detect **suspicious financial transaction patterns** in real time using **Big Data and machine learning techniques**. It collects large volumes of transaction data from different banking sources, including information such as transaction amount, time, location, and whether the transfer is cross-border. The data is then processed using distributed platforms like **Apache Spark**, which help in handling and analyzing millions of records efficiently and quickly.

Once the data is cleaned and prepared, the system applies **unsupervised learning algorithms** like **DBSCAN clustering** to identify unusual or suspicious behavior. This helps in detecting patterns that do not follow normal transaction trends — for example, sudden large transfers, frequent small transactions, or abnormal cross-border activity. These clusters of anomalies are then flagged for further investigation by compliance teams.

The overall system works in a **real-time environment**, where streaming data from financial transactions is continuously monitored. Using **Big Data tools and cloud platforms**, the system ensures scalability, high speed, and accuracy in detecting potential money laundering activities. This not only improves the efficiency of financial monitoring but also supports banks and institutions in meeting regulatory compliance and preventing financial crimes.

#### System Architecture

The overall **Big Data architecture** for the Energy Theft Detection System consists of the following major components:

**Step-by-Step Flow:**

**1.Data Collection Layer:**

Transaction data is collected from multiple banking sources, including details such as transaction ID, time, amount, city, counterparty country, and cross-border status. The data may come from live transaction streams or stored logs and is transferred into the Big Data environment using tools like **Apache Kafka** or **Flume**.

**2.Data Storage Layer:**

The collected data is stored in a distributed storage system such as **HDFS (Hadoop Distributed File System)** or **Google Cloud Storage (GCS)**. This allows handling of high-volume datasets efficiently and ensures fault tolerance and scalability.

**3.Data Pre-Processing Layer:**

In this stage, the raw data is cleaned and transformed using **Apache Spark**. Missing values are handled, categorical values are encoded, and numerical features are normalized. This prepares the data for machine learning analysis.

**4.Feature Extraction and Selection:**

Relevant attributes such as transaction amount, frequency, cross-border flag, and burst count are extracted. These features help the clustering algorithm identify behavioral patterns among different users or accounts.

**5.Clustering and anomaly detection layer:**

An unsupervised learning algorithm such as **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)** is applied to detect clusters of normal transactions and isolate suspicious ones (outliers). This helps in automatically identifying transactions that deviate from regular patterns.

**6.Real time and alert generation Layer:**

The system continuously monitors live transaction streams. When anomalies are detected, alerts are generated and sent to financial analysts or AML officers for further investigation.

**7.Sytem result and Evaluation Layer:**

The results are visualized using tools like **Power BI**, **Tableau**, or **Spark dashboards**, which display transaction clusters, risk scores, and anomaly patterns. This helps in understanding suspicious behavior trends across users and regions.

### 

### 3.3 System Requirements

#### 3.3.1 Hardware Requirements

#### To implement the Real-Time AML Suspicious Pattern Clustering System effectively, a computer with moderate to high processing capacity is required to handle large datasets and run machine learning models efficiently. The minimum hardware configuration includes an Intel Core i5 processor or higher (or equivalent AMD processor) to ensure smooth data processing performance. The system should have at least 8 GB of RAM for handling Spark computations, though 16 GB or more is recommended for faster data analytics. A 500 GB hard disk is necessary to store raw data, intermediate results, and model outputs, with an SSD drive preferred for quicker read/write operations. The system should also support a 64-bit operating system and have a stable internet connection to enable cloud-based processing and access to Big Data platforms like Google Cloud or AWS. Additionally, a GPU-enabled system is optional but beneficial for large-scale or high-speed computations.

#### 3.3.2 Software Requirements

The Real-Time AML Suspicious Pattern Clustering System requires a robust software environment capable of handling Big Data processing and machine learning tasks. The project is implemented using Apache Spark for distributed data processing and Python as the main programming language. Key Python libraries such as pandas, NumPy, scikit-learn, and matplotlib are used for data manipulation, clustering, and visualization. The system runs on a 64-bit operating system like Windows 10/11, Linux (Ubuntu), or macOS. For managing and storing data, Hadoop (HDFS) or Google Cloud Storage (GCS) is utilized. The development and execution environment includes Jupyter Notebook or Google Colab for coding, and Power BI or Tableau for visualizing suspicious transaction clusters. Cloud platforms such as Google Cloud Dataproc or AWS EMR can be integrated to provide scalability and real-time analytics capabilities

* 1. **System Modules**

**The Real-Time AML Suspicious Pattern Clustering System built on Databricks consists of several interconnected modules that work together to process financial transaction data, detect anomalies, and visualize results efficiently.**

**1. Data Ingestion Module**

In this module, transaction data from multiple banking sources is imported into Databricks using Databricks File System (DBFS) or connectors like Azure Data Lake, AWS S3, or Google Cloud Storage. The data includes columns such as event\_id, event\_time, city, amount, is\_cross\_border, and aml\_suspect. Databricks automatically manages scaling and distributed ingestion for large datasets.

**2. Data Preprocessing Module**

The ingested data is cleaned and transformed using PySpark DataFrames within Databricks. Missing values are handled, data types are standardized, and categorical variables like channel or counterparty\_country are encoded. This ensures that the data is structured and ready for analysis. You can also use Delta Lake for maintaining data quality and version control**.**

**3. Feature Engineering Module**

In this step, important features such as transaction amount patterns, cross-border activity, and burst\_count\_24h are extracted. New derived features (e.g., total transactions per day, transaction ratios) can also be created using PySpark SQL or Databricks notebooks to enhance model accuracy.

**4. Clustering & Anomaly Detection Module**

This is the core analytical module where the DBSCAN algorithm (implemented through scikit-learn or MLlib) is applied. The algorithm identifies dense regions of normal transactions and separates outliers as suspicious ones. These anomalies may indicate potential money laundering activities or fraudulent behaviors.

**5. Real-Time Processing & Alert Module**

Using Databricks Structured Streaming, the system processes live incoming transactions in real time. Whenever unusual patterns or outliers are detected, alerts are triggered and stored in a separate Delta table for review by AML analysts.

**6. Visualization & Reporting Module**

The output clusters and anomalies are visualized using Databricks’ built-in dashboards or external tools like Power BI or Tableau connected through JDBC/ODBC. Interactive visuals display transaction trends, high-risk clusters, and suspicious activity across different regions or accounts.

**7. Cloud Integration & Scalability Module**

Since Databricks runs on cloud platforms (like Azure Databricks, AWS Databricks, or GCP Databricks), it ensures auto-scaling, high availability, and parallel processing. This makes the system suitable for large-scale real-time AML monitoring in financial institutions.

#### Data Flow Diagram (DFD) Level 0 DFD:

User → Banking Transaction System → Databricks File System (DBFS) → PySpark (Preprocessing & Feature Engineering) → MLlib / Scikit-learn (DBSCAN Clustering) → Delta Table → Power BI / Databricks Dashboard → Visualization Output

#### Level 1 DFD:

#### Level 1 DFD:

#### ┌────────────────────────────┐

#### │ Banking Transaction Data │

#### │ (event\_id, amount, city, │

#### │ country, cross\_border etc.)│

#### └──────────────┬──────────────┘

#### │

#### ▼

#### ┌────────────────────────────┐

#### │ 1. Data Collection Module │

#### │ → Databricks File System│

#### │ (DBFS / Cloud Input) │

#### └──────────────┬──────────────┘

#### │

#### ▼

#### ┌────────────────────────────┐

#### │ 2. Data Cleaning & Feature │

#### │ Extraction (PySpark) │

#### │ - Remove nulls/duplicates │

#### │ - Extract features (freq, │

#### │ burst\_count, patterns) │

#### └──────────────┬──────────────┘

#### │

#### ▼

#### ┌────────────────────────────┐

#### │ 3. Processed Data Storage │

#### │ → Delta Tables (Processed│

#### │ Data Repository) │

#### └──────────────┬──────────────┘

#### │

#### ▼

#### ┌────────────────────────────┐

#### │ 4. Clustering & Anomaly │

#### │ Detection (DBSCAN via │

#### │ MLlib / scikit-learn) │

#### │ - Identify clusters │

#### │ - Detect suspicious outliers│

#### └──────────────┬──────────────┘

#### │

#### ▼

#### ┌────────────────────────────┐

#### │ 5. Visualization & Report │

#### │ → Power BI / Databricks │

#### │ Dashboard (Anomalies, │

#### │ Risk Clusters) │

#### └────────────────────────────┘

#### Summary

The proposed system is designed to detect and analyze anomalies in large-scale datasets using machine learning and big data techniques on the Databricks platform. It efficiently processes vast amounts of data, performs real-time analysis, and identifies unusual patterns that may indicate potential risks or abnormal activities. The system integrates modules for data collection, preprocessing, clustering-based anomaly detection, and visualization of results. By leveraging the power of cloud and AI, it ensures high accuracy, scalability, and faster insights, making it suitable for various real-world applications like fraud detection, safety monitoring, and predictive maintenance.

### CHAPTER 4

### MODULES DESCRIPTION

#### 4.1 Data Collection Module

#### The Data Collection Module is the first and most crucial part of the system, responsible for gathering raw data from various sources such as sensors, logs, databases, and real-time streaming platforms. In the Databricks environment, this data can be imported from cloud storage services like AWS S3, Azure Blob, or Google Cloud Storage, or even connected APIs. The collected data may include both structured and unstructured formats such as CSV, JSON, or Excel files.

#### This module ensures data consistency and completeness by validating input sources and handling missing or corrupted records. It uses Databricks notebooks and PySpark for data ingestion and schema definition, allowing the system to handle large-scale datasets efficiently.

#### Overall, the Data Collection Module lays the foundation for the entire pipeline by ensuring that high-quality and relevant data is continuously gathered for further preprocessing, analysis, and anomaly detection.

#### 

#### 4.2 Data Preprocessing Module

The **Data Preprocessing Module** plays a vital role in preparing the raw data for effective analysis and model training. In this module, the collected data is cleaned by removing duplicates, handling missing values, and correcting inconsistencies. Using **Databricks and PySpark**, the data is transformed into a structured format through normalization, feature selection, and encoding of categorical variables. Outliers are detected and managed to improve data reliability. This module ensures that the dataset is consistent, accurate, and ready for the next stages such as feature extraction, visualization, and model development, thereby enhancing the overall performance of the system.

#### 

#### 4.3 Hive Query & Analysis Module

Hive is used to perform **structured analysis on large-scale datasets**.

**1. Daily Transaction Volume**

SELECT date(event\_time) AS txn\_date, COUNT(\*) AS total\_txns

FROM transactions

GROUP BY date(event\_time)

ORDER BY txn\_date;

**2. Daily Suspicious Transaction Trend**

SELECT date(event\_time) AS txn\_date, COUNT(\*) AS suspicious\_txns

FROM transactions

WHERE aml\_suspect = TRUE

GROUP BY date(event\_time)

ORDER BY txn\_date;

**3. Top 10 Cities by Transaction Volume**

SELECT city, COUNT(\*) AS txn\_count

FROM transactions

GROUP BY city

ORDER BY txn\_count DESC

LIMIT 10;

**4. Top 10 Cities by Suspicious Transactions**

SELECT city, COUNT(\*) AS suspicious\_txns

FROM transactions

WHERE aml\_suspect = TRUE

GROUP BY city

ORDER BY suspicious\_txns DESC

LIMIT 10;

**5. Cross-Border Transaction Split**

SELECT is\_cross\_border, COUNT(\*) AS txn\_count

FROM transactions

GROUP BY is\_cross\_border;

**6. Average Transaction Amount per Channel**

SELECT channel, ROUND(AVG(amount), 2) AS avg\_amount

FROM transactions

GROUP BY channel;

**7. Suspicious Transactions by Channel**

SELECT channel, COUNT(\*) AS suspicious\_txns

FROM transactions

WHERE aml\_suspect = TRUE

GROUP BY channel

ORDER BY suspicious\_txns DESC;

**8. Cross-Border Transactions by Country**

SELECT counterparty\_country, COUNT(\*) AS txn\_count

FROM transactions

WHERE is\_cross\_border = TRUE

GROUP BY counterparty\_country

ORDER BY txn\_count DESC;

**9.Top 10 High-Value Accounts**

SELECT account\_id, ROUND(SUM(amount), 2) AS total\_amount

FROM transactions

GROUP BY account\_id

ORDER BY total\_amount DESC

LIMIT 10;

#### 4.4 Visualization Module

The **Visualization Module** helps present the results of the AML suspicious pattern detection process in a clear and easy-to-understand way. After applying the **DBSCAN algorithm**, this module displays the clusters of normal transactions and highlights the outliers or suspicious ones using charts and graphs. It uses Databricks notebooks to create visualizations like scatter plots, heatmaps, and dashboards that show transaction patterns, frequency, and anomaly distributions. This makes it easier for analysts and banking officials to identify unusual behaviors quickly, understand relationships between data points, and make informed decisions for further investigation or action.

#### 

#### 4.5 Dashboard Module

The **Dashboard Module** is designed to visually represent key insights extracted from transactional data using SQL queries executed within **Databricks**. It provides an interactive platform to monitor transaction behavior, detect anomalies, and track suspicious patterns. The queries calculate important metrics such as **daily transaction volume**, **number of suspicious transactions per day**, **top cities by transaction count**, **average transaction amounts by channel**, and **suspicious transaction rates** across different regions. They also analyze **cross-border activities**, **burst transactions within 24 hours**, and **account-level total amounts** to uncover hidden trends. Additionally, the output of the **DBSCAN clustering model** is visualized to show **cluster sizes, average amounts, and burst patterns**. All these insights are displayed through **dynamic charts and dashboards** in Databricks or integrated tools like **Power BI**, helping analysts quickly identify risk areas, monitor system performance, and support real-time anti-money laundering decisions.

### CHAPTER 5

### IMPLEMENTATION

### The system implementation begins with setting up the Databricks workspace, where all the data processing and machine learning tasks are performed. Databricks provides a powerful big data environment that allows the handling of large-scale financial transaction data efficiently. The dataset is uploaded to Databricks, which may include details such as transaction ID, account number, amount, city, channel, and transaction time. Using Databricks ensures that data is processed quickly and stored securely in the cloud.

### In the next step, data preprocessing is carried out using PySpark. This involves cleaning the dataset by removing missing or duplicate records, converting text values into numerical formats, and normalizing the data to ensure that all features are on the same scale. The preprocessing stage also includes generating new features such as transaction frequency and deviation from average spending to help detect unusual patterns more effectively.

### After preprocessing, the DBSCAN algorithm is implemented to detect anomalies in the financial transactions. DBSCAN groups together transactions that have similar spending patterns and identifies points that do not belong to any cluster as anomalies or suspicious activities. Since DBSCAN does not require the number of clusters to be specified in advance, it is well-suited for real-world transaction data where patterns are not always known. The model helps uncover hidden patterns of fraudulent behavior without prior labeling.

### Finally, the results are visualized and evaluated within Databricks. The clusters formed by DBSCAN are displayed using scatter plots, and anomalies are highlighted for better understanding. Evaluation metrics such as silhouette score and cluster density are analyzed to ensure model accuracy. This implementation helps build an automated, data-driven system capable of identifying financial irregularities using big data and clustering techniques, improving fraud detection efficiency and accuracy.

### CHAPTER 6

### RESULTS AND DISCUSSION

The results of the system demonstrate **highly actionable insights into urban crime patterns**. For example, the dashboard highlights that In India, **Chennai and Delhi consistently rank as the cities with the highest total crimes**, with certain neighborhoods recurring in the top 3 hotspots. The **top 20 hotspot map** visualizes all critical areas, allowing authorities to prioritize **patrolling and resource allocation** effectively.

**Energy theft Crime Summary:** Pie charts show the percentage distribution of crime types per city. For instance, Chennai’s dataset may reveal that **theft and assault constitute over 60% of reported crimes**, while burglary accounts for 20%. These visualizations help authorities focus on **preventive measures specific to dominant crime types**.

**Top 3 Hotspots per City:** By dividing each city into grids, we can see **localized crime intensity patterns**. In Delhi, grids covering areas such as Connaught Place, Karol Bagh, and Dwarka West are consistently high in crime. This information supports **focused deployment of patrols, CCTV** installation, and community awareness programs.

**Temporal Analysis:** Bar and line charts illustrate **monthly and seasonal crime trends**. Data often shows peaks during specific months, e.g., festivals or holiday seasons, highlighting the need for **temporary surge deployments**. Temporal patterns also provide insights for **predictive policing models**, which can forecast crime probabilities for upcoming months.

### RESULTS:

### 

### 

**Figure 2: Crime Hotspots Analysis Dashboard**

The visualization results from the dashboard provide a comprehensive overview of transaction behavior and suspicious activity patterns detected within the dataset. The time-based visualizations — such as total transactions and suspicious transactions by date — help analysts identify **temporal trends**, revealing whether certain days experience unusual spikes in suspicious activity. Similarly, the **hourly transaction distribution** highlights peak transaction periods, enabling the detection of irregular activities that occur during off-peak hours, which may indicate deliberate attempts to avoid monitoring.

City-wise and country-based visualizations offer valuable **geographical insights**, identifying high-risk regions where money laundering or fraudulent activities are more prevalent. For example, charts showing the top cities by suspicious transaction rates or average burst counts reveal potential hotspots of financial irregularities. The **cross-border transaction and channel-based analyses** further allow investigators to understand which transaction types or channels (e.g., online, ATM, or branch) are most commonly associated with suspicious patterns.

Finally, the clustering results derived from the **DBSCAN algorithm** provide a deeper layer of intelligence by grouping similar transactions based on factors like transaction amount and burst frequency. This enables the identification of **hidden anomalies or emerging fraud networks** that traditional rule-based systems might overlook. Together, these visuals transform raw data into meaningful insights, helping compliance teams take proactive measures, optimize fraud detection strategies, and strengthen anti-money laundering defenses through data-driven decisions.

### CHAPTER 7

### CONCLUSION

In conclusion, the **Real-Time AML Suspicious Pattern Clustering System** represents a significant advancement in detecting and preventing fraudulent activities within the banking and financial sectors. The project efficiently leverages the **DBSCAN clustering algorithm** on the **Databricks big data platform** to automatically identify abnormal transaction behaviors that may indicate money laundering or other illicit financial activities. By analyzing multiple parameters such as transaction amount, frequency, time, location, and cross-border involvement, the system uncovers complex and hidden patterns that traditional rule-based methods often fail to detect.

Unlike conventional systems that depend heavily on predefined thresholds or manual verification, this model uses an unsupervised learning approach that adapts dynamically to new and evolving fraud behaviors. The integration of **PySpark** within the Databricks environment enables large-scale parallel data processing, making it possible to analyze millions of records in real time with high speed and accuracy. This significantly reduces the time required for fraud detection and enhances the reliability of the insights produced.

The **DBSCAN algorithm** excels in identifying dense clusters of normal transactions while marking sparse, isolated points as potential anomalies or suspicious activities. This makes it particularly useful for AML (Anti-Money Laundering) scenarios, where fraudulent behaviors are often rare but highly impactful. The system’s ability to handle noisy, irregular, and large-volume financial data ensures that it remains robust and effective even under complex operational conditions.

Furthermore, the visualization of results within Databricks allows analysts to interpret the clustering patterns easily and make informed decisions based on data-driven insights. The outputs can be further integrated into dashboards or alerting systems to notify compliance teams instantly about high-risk transactions. This proactive approach enhances real-time monitoring capabilities and reduces the financial and reputational risks associated with money laundering.

Overall, the developed system combines the power of **machine learning, big data analytics, and cloud computing** to deliver a highly scalable, automated, and intelligent solution for AML detection. It not only improves operational efficiency but also supports regulatory compliance by providing transparent and traceable data-driven results. By implementing this project, financial institutions can move closer to achieving a secure, efficient, and trustworthy banking environment that is resilient against fraudulent and suspicious activities.

### CHAPTER 8

### FUTURE ENHANCEMENTS

In the future, the Real-Time AML Suspicious Pattern Clustering System can be enhanced in several ways to improve its accuracy, adaptability, and real-world applicability. One major improvement would be integrating advanced deep learning models such as Autoencoders or Graph Neural Networks (GNNs) to capture more complex relationships between transactions and accounts, enabling the detection of highly sophisticated money laundering patterns. Additionally, incorporating real-time streaming data from multiple financial sources using tools like Apache Kafka or Spark Streaming would make the system capable of detecting anomalies as they occur, ensuring faster response times for potential threats.

Another enhancement could involve combining multiple clustering and anomaly detection algorithms, such as Isolation Forest, LOF (Local Outlier Factor), or K-Means Hybrid Models, to create an ensemble-based detection mechanism. This hybrid approach would increase the robustness of the system and reduce false positives, providing more accurate insights. Moreover, integrating Natural Language Processing (NLP) techniques could help analyze unstructured data sources like transaction descriptions or customer communications for early fraud indicators.

In terms of scalability and deployment, the system can be extended to a cloud-native microservices architecture, enabling easy integration with banking APIs and regulatory platforms. A user-friendly dashboard or web portal can also be developed for compliance officers to visualize clusters, review suspicious transactions, and generate automated reports. Future versions could also include explainable AI (XAI) techniques to make model decisions more transparent and understandable to auditors and regulators.

Finally, implementing a feedback learning mechanism where the system continuously learns from confirmed suspicious cases can make the model self-improving over time. By integrating such intelligent automation and real-time feedback, the project could evolve into a fully autonomous, adaptive, and regulatory-compliant AML monitoring platform, setting a strong benchmark for future financial crime detection systems.

### REFERENCES

**[1] Apache Hive Documentation** – “Apache Hive – Data Warehousing and SQL-Like Queries.” [Online]. Available: <https://cwiki.apache.org/confluence/display/Hive/Home>

**[2] Apache Hadoop Documentation** – “Hadoop: Distributed Storage and Processing Framework.” [Online]. Available: https://hadoop.apache.org/

**[3] Folium Python Library Documentation** – “Folium – Python Data Visualization for Maps.” [Online]. Available: [https://python-](https://python-visualization.github.io/folium/) [visualization.github.io/folium/](https://python-visualization.github.io/folium/)

**[4] Chart.js Documentation** – “Chart.js – Open Source JavaScript Charts.” [Online]. Available: <https://www.chartjs.org/>

**[5] Government of India Open Data Portal** – “Crime India Dataset.” [Online]. Available: https://data.gov.in/catalog/crime-india

**[6]** R. S. Brunsdon and L. Comber, *An Introduction to GIS and Spatial Analysis in Criminology*, 2nd ed., Routledge,