SIMATS ENGINEERING

SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES – 602105

A CAPSTONE PROJECT REPORT

CSA1583-Cloud computing and Big Data Analytics Using Cloud Federation

Submitted in the partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

Abnormal Trading System

A PROJECT REPORT

Done by:

B.Tejaswi (192210670)

Supervisor

Dr.Chenni Kumaran

DECLARATION

I am B.Tejaswi, students of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, hearby declare that the work presented in the Capstone project entitled the outcome of our own bonafide work and is undertaken care of Engineering Ethics.

B.Tejaswi(192210670)

Date:

Day:

CERTIFICATE

This is to certify that the project entitled "Abnormal Trading System Using Open Cloud Platform submitted by B.Tejaswi has been carried out under our supervision. The project has been submitted as per the requirements in the current semester of B. Tech Information Technology.

Faculty-in-Charge

Dr.Chenni Kumaran

Table of Contents

SI. NO	Components	Marks(100)
1	**Problem Statement**	**(5)**
2	**Requirement Analysis**	**(10)**
	• Identifying specific requirments	(5)
	• Determine the necessary features	(5)

3	**Architecture Design**	**(5)**
4	**Data Ingestion and Processing**	**(10)**
5	**Anomaly detection**	**(10)**
6	**Real-time Monitoring and Altering**	**(10)**
7	**Visualization and Reporting**	**(20)**
	• **a. Design interactive dashboards**	(10)
	• **b. Use cloud native analysis and services**	(10)
8	**Cloud integration and development**	**(10)**
9	**Testing**	**(5)**
10	**Performance evaluation**	**(10)**
11	**Conclusion**	**(5)**
	Total Marks	**(100)**

ABNORMAL TRADING SYSTEM

Abstract: This project focuses on Abnormal Trading System. In the realm of financial markets, detecting abnormal trading activities has become crucial for maintaining market integrity and investor confidence. This paper proposes an advanced abnormal trading system designed to identify suspicious patterns and behaviors in trading data.

The system integrates machine learning algorithms with statistical analysis to dynamically assess deviations from normal market behavior. Key features include real-time data processing, anomaly detection using clustering techniques, and predictive modeling to anticipate potential market disruptions. Practical applications encompass regulatory compliance, risk management, and early warning systems for financial institutions. The effectiveness of the system is validated through historical data analysis and case studies, demonstrating its capability to detect anomalies with high accuracy and efficiency. This research contributes to enhancing market surveillance capabilities and fostering a more transparent and secure trading environment.

Introduction: The global financial markets are increasingly complex and interconnected, providing fertile ground for various forms of misconduct, including market manipulation and insider trading. Detecting and mitigating these activities is not only essential for maintaining market integrity but also crucial for ensuring investor confidence and regulatory compliance. An abnormal trading system plays a pivotal role in this landscape, serving as a sophisticated tool designed to identify irregular trading patterns and behaviors that deviate significantly from established norms. The need for such systems has grown exponentially with the rise of high-frequency trading and algorithmic trading strategies, which can execute trades at speeds and volumes that challenge traditional surveillance methods. These advancements, while enhancing market efficiency, also create vulnerabilities that can be exploited by malicious actors. Therefore, there is a pressing demand for robust technological solutions capable of continuously monitoring market data in real-time, detecting anomalies, and alerting relevant authorities promptly.

This paper explores the development and implementation of an abnormal trading system that leverages cutting-edge technologies such as machine learning, artificial intelligence, and advanced statistical analysis. By harnessing these tools, the system aims to not only detect but also predict abnormal trading behaviors with a high degree of accuracy and efficiency. The integration of predictive modeling techniques enables proactive risk management and preemptive action, thereby reducing the potential impact of abnormal trading activities on market stability.

Problem Statement:

Developing a comprehensive Abnormal Trading System to streamline operations, enhance customer service, and optimize efficiency poses several challenges. The rapid evolution of trading technologies, including high-frequency trading and complex algorithms, has exacerbated vulnerabilities in financial markets, leading to increased instances of market manipulation, insider trading, and other illicit activities. Traditional surveillance methods often struggle to detect these abnormal trading behaviors in real-time, posing significant challenges to market integrity and regulatory compliance. There is a critical need for an advanced abnormal trading system capable of accurately and promptly identifying irregular trading patterns and behaviors.

Proposed Design Work:

Requirement Analysis:

1. Identifying Key Components:

Market Data Feed:

A reliable source of real-time and historical market data, including price quotes, volumes, bid/ask spreads, and other relevant trading information.

2. Order Management System (OMS):

Handles the entire lifecycle of orders placed by traders, including order routing, execution, allocation, and confirmation.

3. Execution Management System (EMS):

Facilitates the efficient execution of trades across multiple venues or exchanges, often integrating with the OMS for seamless order handling.

4. Risk Management Tools:

Implements risk controls to monitor and manage exposure to market, credit, and operational risks, ensuring compliance with regulatory requirements and internal risk limits.

5. Algorithmic Trading Strategies:

Automated trading algorithms designed to execute orders based on predefined rules or parameters, optimizing trade execution and reducing market impact.

6. Back-Testing and Simulation:

Tools to test trading strategies using historical data to assess performance metrics, refine algorithms, and validate trading ideas before deployment.

7. Reporting and Analytics:

Generates performance reports, trade analytics, and compliance reports to evaluate trading activities, monitor performance, and ensure regulatory compliance.

8. Scalability and Flexibility:

Designed to handle increasing trading volumes, diverse asset classes, and evolving market conditions, while remaining adaptable to technological advancements and regulatory changes.

Features:

Real-Time Monitoring:

Continuously monitors market data in real-time to identify abnormal trading patterns, deviations from historical norms, or suspicious behaviors

Anomaly Detection:

Utilizes advanced statistical analysis, machine learning algorithms, and pattern recognition techniques to detect anomalies in trading data that may indicate market manipulation, insider trading, or other illicit activities.

Investigation Support:

Provides tools and functionalities to conduct detailed investigations into flagged anomalies, including access to historical trading data, transaction records, and market participant information.

Compliance Monitoring:

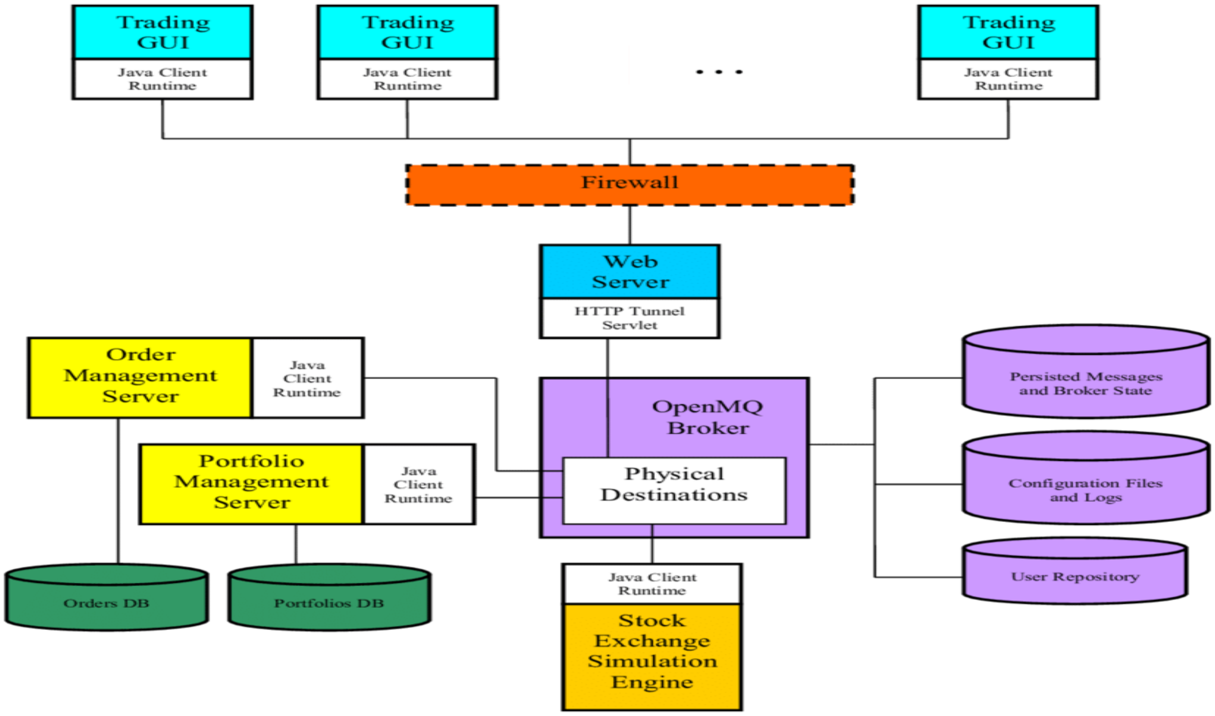
Ensures adherence to regulatory requirements by monitoring trading activities for compliance with market rules, insider trading regulations, and other regulatory standards.

Risk Management:

Assesses and manages risks associated with abnormal trading activities, including potential market disruptions, financial losses, and reputational risks for market participants.

Architectural Design

The architectural design of an Abnormal Trading System is structured to detect, analyze, and respond to irregular trading activities within financial markets effectively. At its core, the system begins with a robust data ingestion layer that aggregates real-time market data feeds and transaction records from diverse sources. This data is then processed through a series of normalization and preprocessing steps to ensure consistency and accuracy. The heart of the system lies in its anomaly detection and pattern recognition capabilities, which utilize advanced Integration with existing trading infrastructure such as Order Management Systems (OMS) and Execution Management Systems (EMS) ensures coordinated responses and compliance with regulatory standards. Security measures are paramount, ensuring data integrity and confidentiality, while scalability features allow the system to handle increasing volumes of data and adapt to evolving market conditions. Overall, this architecture design aims to enhance market surveillance, mitigate risks associated with abnormal trading activities, and maintain the integrity and transparency of financial markets.



Data Ingestion and Processing:

Data ingestion and processing for an abnormal trading system involves collecting real-time and historical market data from various sources such as stock exchanges, financial news feeds, and social media platforms. This data is then cleaned and pre-processed to remove any inconsistencies, duplicates, and irrelevant information.

Advanced algorithms and machine learning models analyze the processed data to detect patterns and anomalies indicative of abnormal trading activities, such as insider trading or market manipulation. The system continuously monitors and updates its models to adapt to new market conditions, ensuring accurate and timely identification of suspicious activities. Finally, alerts are generated and sent to regulatory bodies or compliance teams for further investigation and action.

Anomaly Detection:

Anomaly detection in an abnormal trading system involves using advanced statistical methods and machine learning algorithms to identify unusual patterns or behaviors in trading data. The system ingests and processes real-time and historical data from various sources, including stock exchanges, financial news, and social media. Pre-processing steps, such as normalization, outlier removal, and feature extraction, are applied to ensure data quality and relevance. Machine learning models, including supervised, unsupervised, and semi-supervised learning techniques, are trained on this data to detect deviations from normal trading patterns.

Techniques such as clustering, neural networks, and time-series analysis help identify anomalies like unusual trading volumes, rapid price changes, or irregular trading times. Continuous monitoring and model updates ensure the system adapts to evolving market conditions, improving its accuracy and effectiveness in detecting potential market abuse. Detected anomalies trigger alerts for regulatory bodies or compliance teams to investigate and take appropriate action.

Real-time Monitoring and Altering:

Data Ingestion: Continuously collect real-time trading data from multiple sources such as stock exchanges, financial news feeds, and social media platforms.

- **Response and Investigation**: Enable stakeholders to quickly review and investigate alerts, providing tools and interfaces to analyze the suspicious activities and determine the appropriate course of action.

- **Feedback Loop**: Incorporate feedback from investigations to continuously improve the detection models, updating them with new patterns and insights to enhance accuracy.

- **Scalability and Performance**: Ensure the system is scalable to handle high volumes of data and capable of performing real-time processing without significant latency.

- **Thresholds and Rules**: Define thresholds and rules for what constitutes abnormal trading behavior, adjusting them dynamically based on market conditions and model feedback.

- **Alert Generation**: Automatically generate alerts when the system detects anomalies that exceed predefined thresholds. These alerts should include detailed information about the detected anomaly, such as time, involved securities, and nature of the anomaly.

Visualization and Reporting:

a. Design interactive dashboards:

Overview Dashboard

- **Market Summary**: Display real-time market metrics such as total trading volume, number of trades, and overall market movement.
- **Anomaly Summary**: Highlight key statistics on detected anomalies, including the number of alerts generated, types of anomalies, and affected securities.
- **Alerts Overview**: Show a summary of the most recent alerts with brief details.

Real-Time Monitoring Dashboard

- **Live Data Feed**: Stream real-time trading data with options to filter by specific securities, exchanges, or time periods.
- **Anomaly Detection**: Visualize detected anomalies in real-time, using heatmaps or highlighted trading activity graphs to indicate unusual behavior.
- **Alert Log**: Maintain a live log of alerts, sortable and searchable by time, type, and severity.

b. Use Cloud-native analytics and visualization services:

Utilizing cloud-native analytics and visualization services for real-time monitoring and alerting in an abnormal trading system offers several advantages, including scalability, flexibility, and ease of integration.

- **Set Up Data Pipelines**: Configure data ingestion pipelines using Kinesis, Pub/Sub, or Event Hubs to stream trading data into your cloud environment.
- **Data Processing**: Implement ETL workflows with Glue, Dataflow, or Data Factory to clean and prepare the data.
- **Deploy ML Models**: Use Sage Maker, AI Platform, or Azure ML to train and deploy your anomaly detection models.

- **Real-Time Processing**: Set up real-time analytics with Kinesis Analytics, Big Query, or

Cloud Integration and Development:

Integrating and deploying a trading system in the cloud involves several steps to ensure that the system is scalable, reliable, and secure. Here’s a comprehensive guide to achieve this using popular cloud platforms:

- **Data Storage**:
  - Store raw data in S3 and processed data in RDS.
  - Use Redshift for data warehousing.
- **Monitoring and Alerting**:
  - Configure CloudWatch to monitor system metrics.
  - Set up SNS for alert notifications.
- **Visualization**:
  - Build dashboards in QuickSight to visualize trading activities and anomalies.
- **Security**:
  - Manage access with IAM.
  - Encrypt data using KMS.

Testing:

Testing an abnormal trading system involves various phases and methodologies to ensure that the system works as expected, detects anomalies accurately, and performs well under different conditions.

1.Unit Testing

- **Component Testing**: Write unit tests for individual components, such as data ingestion modules, data processing functions, and anomaly detection algorithms.
- **Test Cases**: Ensure test cases cover typical scenarios, edge cases, and potential errors.
- **Mocking**: Use mocking frameworks to simulate external dependencies and services.

2. Integration Testing

- **End-to-End Data Flow**: Test the entire data flow from ingestion to alert generation to ensure components work together seamlessly.
- **Data Consistency**: Verify that data is correctly transformed and remains consistent throughout the pipeline.
- **Service Interaction**: Ensure that different services (e.g., data storage, machine learning models, alerting mechanisms) interact correctly.

3. Functional Testing

- **Anomaly Detection Accuracy**: Evaluate the system's ability to detect true positives (real anomalies) and minimize false positives (incorrectly flagged anomalies).
- **Scenario-Based Testing**: Create test scenarios that mimic real trading activities, including normal trading patterns and known abnormal behaviors.
- **Thresholds and Rules**: Test the effectiveness of predefined thresholds and rules for triggering alerts.

4. Performance Testing

- **Load Testing**: Simulate high-volume trading data to ensure the system can handle peak loads without degradation in performance.
- **Latency Testing**: Measure the time taken for data to flow through the system and for anomalies to be detected and alerts generated.
- **Scalability Testing**: Test the system's ability to scale horizontally and vertically to accommodate growing data volumes and user load.

Performance Evaluation:

Performance evaluation of a trading system, particularly one designed for detecting abnormal trading activities, is crucial to ensure it operates efficiently and meets the required standards. Here's a detailed approach to performance evaluation:

Accuracy Evaluation

- **Precision and Recall**: Calculate precision (the ratio of true positive alerts to total positive alerts) and recall (the ratio of true positive alerts to all actual anomalies).
- **Confusion Matrix**: Use a confusion matrix to evaluate true positives, true negatives, false positives, and false negatives.
- **ROC Curve and AUC**: Plot the Receiver Operating Characteristic (ROC) curve and calculate the Area Under the Curve (AUC) to evaluate the model's performance.

1. Scalability Testing

- **Horizontal Scaling**: Test the system's ability to scale horizontally by adding more instances of data processing nodes.
- **Vertical Scaling**: Evaluate the impact of increasing the capacity (CPU, memory) of individual instances.
- **Auto-Scaling**: Test the effectiveness of auto-scaling configurations under varying loads.

2. Reliability Testing

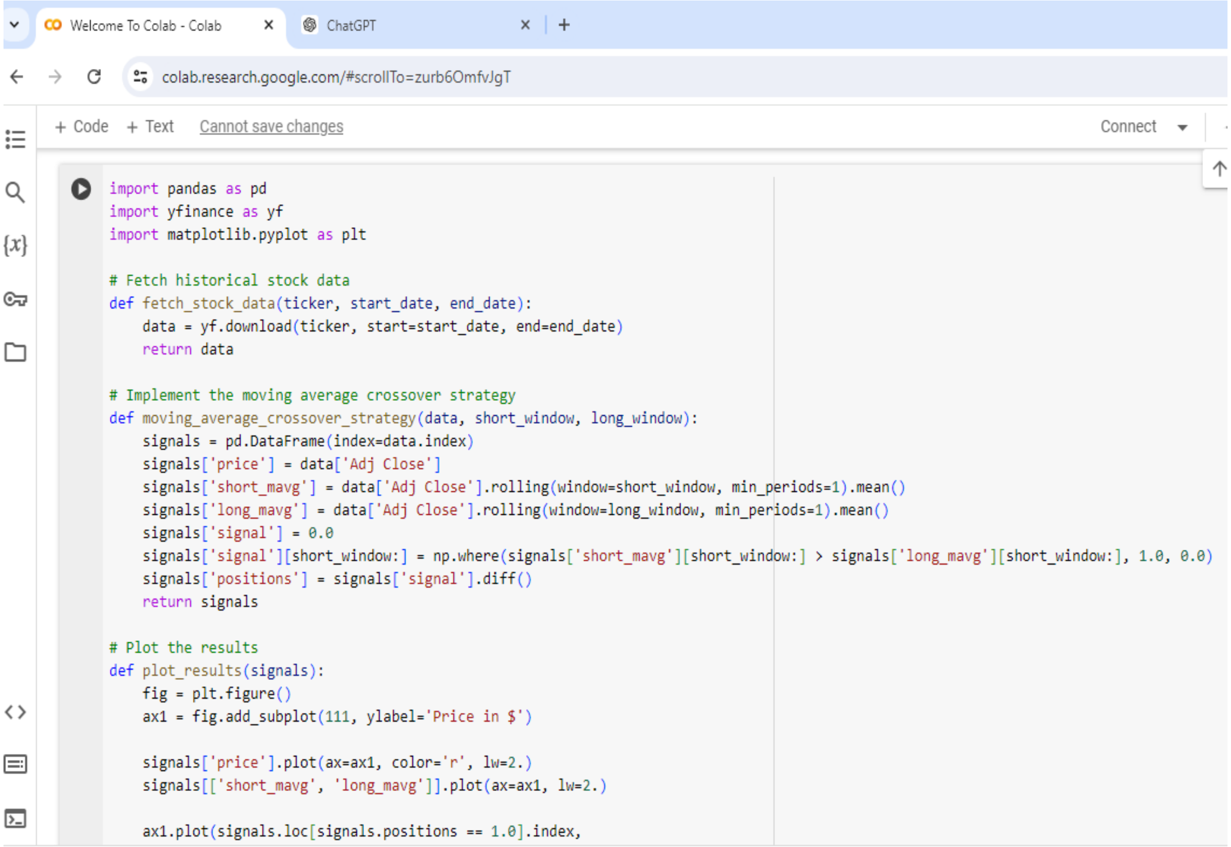
- **Fault Injection**: Simulate failures (e.g., network outages, component crashes) to test the system's resilience and recovery mechanisms.
- **Recovery Time**: Measure the time taken to recover from failures and restore normal operation.
- **Uptime Monitoring**: Use monitoring tools like Amazon CloudWatch, Google Cloud Monitoring, or Azure Monitor to track system uptime and reliability.

3. Resource Utilization

- **Monitoring Tools**: Use tools like Prometheus, Grafana, or built-in cloud monitoring solutions to track CPU, memory, and network usage.
- **Efficiency Analysis**: Analyze resource utilization data to identify inefficiencies and optimize resource allocation.

4.Throughput Testing

- **Load Testing**: Use tools like Apache JMeter, Locust, or Gatling to simulate high volumes of trading data and measure how many transactions the system can handle per second.



```python
import pandas as pd
import yfinance as yf
import matplotlib.pyplot as plt

# Fetch historical stock data
def fetch_stock_data(ticker, start_date, end_date):
    data = yf.download(ticker, start=start_date, end=end_date)
    return data

# Implement the moving average crossover strategy
def moving_average_crossover_strategy(data, short_window, long_window):
    signals = pd.DataFrame(index=data.index)
    signals['price'] = data['Adj Close']
    signals['short_mavg'] = data['Adj Close'].rolling(window=short_window, min_periods=1).mean()
    signals['long_mavg'] = data['Adj Close'].rolling(window=long_window, min_periods=1).mean()
    signals['signal'] = 0.0
    signals['signal'][short_window:] = np.where(signals['short_mavg'][short_window:] > signals['long_mavg'][short_window:], 1.0, 0.0)
    signals['positions'] = signals['signal'].diff()
    return signals

# Plot the results
def plot_results(signals):
    fig = plt.figure()
    ax1 = fig.add_subplot(111, ylabel='Price in $')

    signals['price'].plot(ax=ax1, color='r', lw=2.)
    signals[['short_mavg', 'long_mavg']].plot(ax=ax1, lw=2.)

    ax1.plot(signals.loc[signals.positions == 1.0].index,
```

Peak Load Testing: Test the system's performance under peak trading conditions, which typically occur during market open and close times. **Accuracy Evaluation**

- **Precision and Recall**: Calculate precision (the ratio of true positive alerts to total positive alerts) and recall (the ratio of true positive alerts to all actual anomalies).
- **Confusion Matrix**: Use a confusion matrix to evaluate true positives, true negatives, false positives, and false negatives.

Conclusion:

In conclusion, developing and deploying an abnormal trading system entails integrating sophisticated data ingestion, processing, and anomaly detection capabilities within a robust cloud infrastructure. This system is essential for detecting and responding to irregular trading behaviors promptly, ensuring market integrity and regulatory compliance.

By leveraging advanced machine learning models and real-time monitoring dashboards, stakeholders can gain actionable insights into trading activities, supported by scalable and efficient cloud-native services. Continuous testing and performance evaluation are crucial to optimizing system reliability, accuracy, and responsiveness. Ultimately, such a system not only enhances market security and transparency but also strengthens trust among participants by effectively mitigating risks associated with abnormal trading activities.