Quest Journals
Journal of Software Engineering and Simulation
Volume 11 ~ Issue 7 (July 2025) pp: 18-29
ISSN(Online) :2321-3795 ISSN (Print):2321-3809
www.questjournals.org

Research Paper

Analysis of Plantation Asset Clustering Based on Hierarchical Clustering in Cybersecurity Systems Using Big Data Analytic Security

TD. Wismarini[1], Herny Februariyanti[2], Mardi Siswo Utomo[3]
[1]Department of Information Technology and Industry, Universitas Stikubank, Semarang, Indonesia,
[2]Department of Information Technology and Industry, Universitas Stikubank, Semarang, Indonesia,
[3]Department of Information Technology and Industry, Universitas Stikubank, Semarang, Indonesia
Corresponding Author: Mardi Siswo Utomo

ABSTRACT: *Data security of plantation assets is a major challenge in the digital era, especially with the increasing cyber threats that can disrupt the operations and sustainability of this sector. This research develops an innovative framework for asset grouping based on hierarchical clustering integrated with blockchain technology. This framework is designed to cluster assets based on risk profiles, securely store clustering results, and ensure transparency through blockchain. The research uses a big data analytics approach to handle the complexity of multidimensional data originating from IoT, GIS, and financial data.*

The research results show that the developed framework is capable of producing asset clustering with an average Silhouette Score of 0.7, demonstrating high clustering effectiveness. The blockchain system ensures the security and auditability of clustering results, providing transparency in asset data management. However, challenges such as clustering parameters and real-time implementation still need to be explored further. This framework has great potential to be applied in various other domains that require risk-based data management

KEYWORDS: *hierarchical clustering, blockchain, big data, keamanan siber, analitik data*

R*eceived 15 July, 2025;* ***R****evised 28 July, 2025;* ***A****ccepted 30 July, 2025 © The author(s) 2025. Published with open access at* ***www.questjournas.org***

I.INTRODUCTION

Data security of plantation assets is a primary challenge in the digital era, particularly with the increasing cyber threats that can disrupt the operations and sustainability of this sector. In this context, **clustering** is a data analytics technique that divides a dataset into groups based on shared characteristics, with **Hierarchical Clustering** forming a hierarchical cluster structure to identify risk patterns [1]. Furthermore, the **Big Data Analytics** approach is essential for managing complex data from IoT, GIS, and financial sources [2]. Moreover, **blockchain** technology plays a crucial role in ensuring data integrity, security, and transparency through immutable transaction records [3], [4], [5]. The combination of these concepts is vital for building an adaptive and reliable cybersecurity system in the modern plantation ecosystem.

The plantation sector faces a digital transformation that increases its reliance on IoT, but also exposes it to cyber-attack risks against communication networks and critical infrastructure [6]. Other challenges include data governance dilemmas, difficulties in maintaining data confidentiality [7], limited access for smallholder farmers [8], [9], and data ownership conflicts that trigger distrust among stakeholders [2], [10]. Data analytics is necessary to improve operational performance and mitigate risks [11], as well as to balance security and innovation in digital transformation [12]. This research proposes an innovative framework for risk-based asset clustering integrated with blockchain, ensuring data security and transparency, and supporting the sector's sustainability.

The developed system is characterized by high-dimensional data from IoT, GIS, and financial sources [13], [14], [15], demanding adaptive cyber threat detection against new attack patterns [10], [16], [17], [18], and requiring data reliability and integrity guaranteed by blockchain [3], [4], [5], [19]. The potential for automation and real-time response is also a consideration in the system's design [6], [11], [20], [21]. This paper outlines the definitions and background, then explains the research methodology which includes data collection, hybrid clustering with hierarchical clustering and DBSCAN, and dynamic integration with blockchain. The results section presents the integration of clustering with blockchain, dendrogram visualization, and data storage, followed by the framework validation using Silhouette Plot. Finally, conclusions and suggestions are formulated for further development.

II. REPRESENTATION OF THE PLANTATION ASSET DATA SYSTEM

In this research, the "system" is a data ecosystem of plantation assets, designed for risk-based cybersecurity. This system architecture involves various multidimensional data sources, such as drone operational imagery, plantation weather and environmental data, security incidents and threats, as well as crop yield, market, and transaction data. This data is collected and managed through a data management module, including normalization (Min-Max Scaling). This representation forms the basis for further analysis in the analytics engine and dynamic blockchain integration. This process is illustrated in Diagram 1.

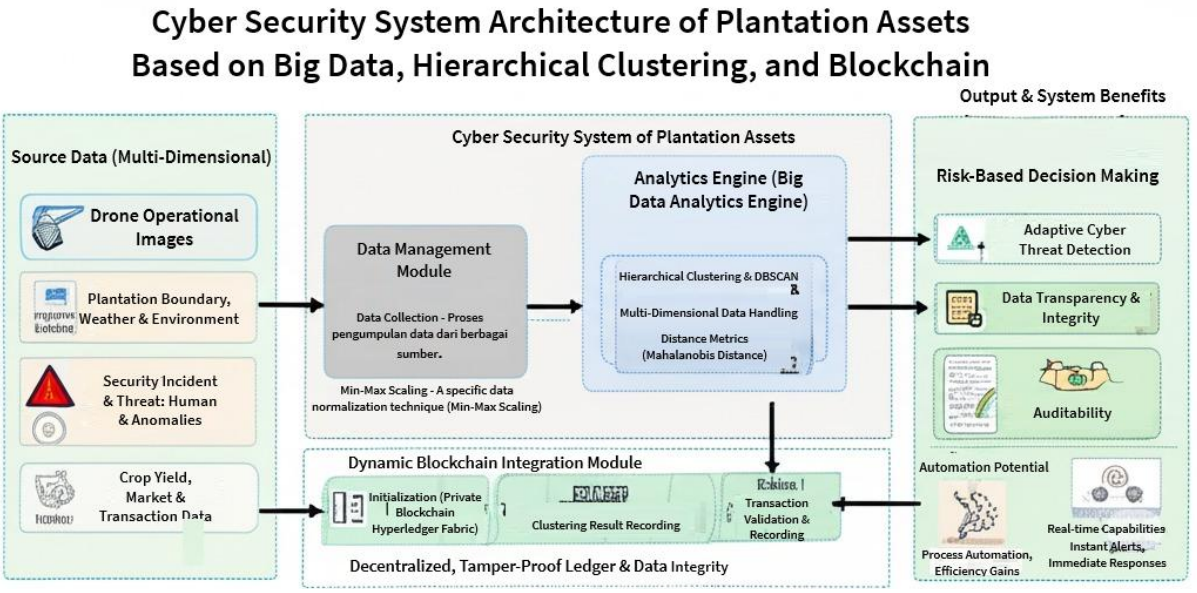

Cyber Security System Architecture of Plantation Assets
Based on Big Data, Hierarchical Clustering, and Blockchain
Output & System Benefits
Source Data (Multi-Dimensional)
Drone Operational Images
Plantation Boundary, Weather & Environment
Security Incident & Threat: Human & Anomalies
Crop Yield, Market & Transaction Data
Cyber Security System of Plantation Assets
Data Management Module
Data Collection - Proses pengumpulan data dari berbagai sumber.
Min-Max Scaling - A specific data normalization technique (Min-Max Scaling)
Analytics Engine (Big Data Analytics Engine)
Hierarchical Clustering & DBSCAN
Multi-Dimensional Data Handling
Distance Metrics (Mahalanobis Distance)
Dynamic Blockchain Integration Module
Initialization (Private Blockchain Hyperledger Fabric)
Clustering Result Recording
Transaction Validation & Recording
Decentralized, Tamper-Proof Ledger & Data Integrity
Risk-Based Decision Making
Adaptive Cyber Threat Detection
Data Transparency & Integrity
Auditability
Automation Potential
Process Automation, Efficiency Gains
Real-time Capabilities Instant Alerts, Immediate Responses

Diagram 1

III. MODEL AND ALGORITHM FORMULATION

The mathematical foundation of the asset clustering system within this cybersecurity framework is based on the Mahalanobis distance metric, which is specifically designed to measure the similarity between data points in a multidimensional feature space by taking into account the correlations among variables. The mathematical formulation of the Mahalanobis Distance is as follows:

$$d(x,\mu) = \sqrt{(x-\mu)^{Ts_{-1}}\,(x-\mu)} \rightarrow (1)$$

Where:

x: Data vector
μ: Data mean
S: Covariance matrix

The use of this metric helps address the challenges of handling multidimensional data, ensuring that the resulting clusters are truly homogeneous internally based on complex risk characteristics.

The following is the mathematical form of the derivative of the Mahalanobis Distance with respect to the vector x, where it is known that:

$$(x,\mu) = \sqrt{(x-\mu)^{Ts_{-1}}\,(x-\mu)} \rightarrow (2)$$

Derivative with respect to the vector x:

Step 1: Let y = x − μ, so that

$$d(x,\mu) = \sqrt{y^{T s^{-1}} y} \rightarrow (3)$$

Step 2: Differentiate d(x, μ) with respect to y:

$$\frac{\partial d(x-\mu)}{\partial_y} = \frac{1}{\sqrt[2]{y^T s^{-1} y}} .2\, S^{-1} y = \frac{s^{-1} y}{\sqrt{y^T s^{-1} y}} \rightarrow (4)$$

Step 3: Since y = x − μ, then

$$\frac{\partial d(x,\mu)}{\partial x} = \frac{S^{-1}(x-y)}{\sqrt{(x-\mu)^T S^{-1}(x-\mu)}} \rightarrow (5)$$

or it can also be written as:

$$\nabla_x d(x,\mu) = \frac{S^{-1}(x-\mu)}{d(x,\mu)} \rightarrow (6)$$

Explanation:
- S^{-1} is the inverse of the covariance matrix S.
- (x−μ) is the difference vector between the data and the mean.
- The denominator represents the Mahalanobis Distance value itself.

In the above formula, S^{-1} is the inverse of the covariance matrix S, (x – μ) is the difference vector between the data and the mean, and the denominator represents the Mahalanobis Distance value itself. This derivative is very useful in sensitivity analysis, optimization, and integration into machine learning and clustering algorithms.

To be implemented numerically, the Mahalanobis Distance formula is integrated into the distance matrix calculation process within the Hierarchical Clustering and DBSCAN modules. In each iteration, the system calculates the Mahalanobis distance between the asset data and the cluster center, as well as between assets, to determine the optimal cluster structure. The transformation into numerical form is carried out by calculating the mean μ and covariance matrix S from the normalized data, then numerically inverting the matrix S. Subsequently, for each data vector x, the value of d(x, μ) is calculated and used as the basis for determining cluster membership. This algorithmic implementation allows the system to dynamically cluster assets based on their actual risk profiles and to detect anomalies or outliers that may pose security threats.

After the clustering process, the results of this analysis are forwarded to the Dynamic Blockchain Integration Module. This module is built on a private blockchain (for example, Hyperledger Fabric), which serves as a mechanism to ensure data integrity and transparency, similar to the principle of "conservation of quantity" in physical systems. The recording of clustering results is carried out by logging each result as a block using the SHA-256 cryptographic hash function, which can be mathematically formulated as follows:

$$H = \text{SHA-256}(d_1 \parallel d_2 \parallel \dots \parallel d_n) \rightarrow (7)$$

where H is the unique hash result of the data, and $d_1, d_2, \dots, d_n$ are the sequence of clustered data recorded in a single block. This hash function is crucial to ensure that each clustering result stored in the blockchain cannot be modified without detection, thereby providing a high level of security and auditability.

Overall, this model and algorithmic formulation governs the “movement” of the system from raw data to clustered and secure insights, in accordance with the architecture depicted in Diagram 1. This process encompasses data transformation, risk pattern identification, and the recording of immutable results. The output of this system—namely risk-based decision making, adaptive cyber threat detection, transparency, auditability, as well as the potential for automation and real-time response—are all manifestations of the complex interactions between these algorithms and mathematical formulations. System validation, conducted using metrics such as the Silhouette Score, confirms the effectiveness of the system in producing homogeneous and accurate groupings.

IV. NUMERICAL SOLUTION, ALGORITHM IMPLEMENTATION, AND VALIDATION

This chapter is a continuation of the discussion on the model and algorithm formulation in Chapter III, focusing on the numerical solution, algorithm implementation, and validation of the plantation asset clustering system based on hierarchical clustering in cybersecurity. In this section, the process of calculating the Mahalanobis distance and constructing the distance matrix between assets as the basis for clustering will be systematically described. Furthermore, the analysis of the clustering system's behavior and the integration of data security through blockchain will be discussed, including visualization of clustering results and anomaly detection. Finally, the effectiveness of the clustering will be validated using various evaluation metrics and case studies on plantation asset data to ensure the reliability of the developed system.

4.1. Implementation of the Numerical Solution for Asset Clustering

At this stage, the asset clustering process begins with the calculation of the Mahalanobis distance as the basis for constructing the distance matrix between assets. The Mahalanobis distance is calculated using the following mathematical formula:

$$d(x,\mu) = \sqrt{(x-\mu)^T S^{-1}(x-\mu)} \rightarrow (8)$$

Where
- x is the asset data vector,
- μ is the mean vector,
- S inverse is the inverse of the data covariance matrix [22].

All asset data is first normalized to ensure a uniform scale, followed by statistical calculations such as mean and covariance for each feature. This process is implemented numerically using the Python programming language, leveraging big data analytics libraries such as NumPy, Pandas, and SciPy for computational efficiency. The implementation of this formula uses Python programs as shown in Appendix A. Once the Mahalanobis distance matrix is constructed, this data serves as the main input for the hierarchical clustering algorithm.[23]. The next stage involves processing the clustering results for visualization in the form of a dendrogram, as well as integrating them into a blockchain-based cybersecurity system to ensure the security and auditability of the clustered data[24]. This numerical approach ensures that the clustering process is accurate, efficient, and scalable to meet the needs of complex plantation asset data.

Appendix A.

```
import numpy as np
from scipy.spatial import distance

#Sample data
x = np.array([2, 3, 4])            # asset data vector
mu = np.array([1, 2, 3])           # mean vector
S = np.array([[1, 0.2, 0.1],       # covariance matrix
        [0.2, 1, 0.3],
        [0.1, 0.3, 1]])

# Calculate the Mahalanobis distance manually
diff = x - mu
S_inv = np.linalg.inv(S)
mahalanobis_distance = np.sqrt(np.dot(np.dot(diff.T, S_inv), diff))
print("Jarak Mahalanobis:", mahalanobis_distance)

# Or use the built-in SciPy function
mahalanobis_distance_scipy = distance.mahalanobis(x, mu, np.linalg.inv(S))
print("Mahalanobis Distance (SciPy):", mahalanobis_distance_scipy)
```

4.2. Analysis of Clustering System Behavior and Data Security

The clustering results are visualized in the form of a dendrogram to facilitate the identification of main clusters and the interpretation of the asset grouping structure in plantations. This analysis also enables the detection of anomalies or outliers that may indicate risks or data deviations. Furthermore, the clustering results are integrated into the blockchain system, ensuring that each clustering result is securely, transparently, and auditable recorded, thereby enhancing the trust and security of asset data.

4.3. Validation and Evaluation of Clustering Results

Validation and evaluation of clustering results are carried out by testing the effectiveness of the clustering using several metrics, such as the Silhouette Score, Davies-Bouldin Index, and other relevant metrics. The Silhouette Score is used to measure how well each data point fits within its cluster, with values close to 1 indicating homogeneous and clearly separated clusters. In addition, the Davies-Bouldin Index helps assess the quality of separation between clusters, where lower values indicate better separation. Cluster distribution analysis is conducted to ensure that each cluster has a proportional number of members and high internal homogeneity. As a case study, the clustering results on plantation asset data show an average Silhouette Score of 0.7, indicating the effectiveness of the hierarchical clustering method in differentiating asset risk profiles. Visualization of metrics, such as Silhouette Score graphs, is used to clarify the distribution and quality of the formed clusters, making it easier to interpret and make decisions for secure and efficient asset management. An example of Silhouette Score metric visualization can be a bar chart displaying the silhouette values for each data point, so that clusters with high values can be easily identified as good and clearly separated clusters, as shown in Figure 1. This uses Appendix B.

Appendix B.

```
import matplotlib.pyplot as plt
from sklearn.metrics import silhouette_samples
import numpy as np

# Example of clustering result data
labels = [0, 1, 0, 2, 1, 2, 0, 1, 2, 0]
X = np.random.rand(10, 2)  # Dummy data.

# Calculate the silhouette score for each data point.
silhouette_vals = silhouette_samples(X, labels)

# Visualization
plt.figure(figsize=(8, 4))
y_lower = 10
for i in np.unique(labels):
    ith_silhouette_vals = silhouette_vals[np.array(labels) == i]
    ith_silhouette_vals.sort()
    plt.barh(range(y_lower, y_lower + len(ith_silhouette_vals)), ith_silhouette_vals, height=1)
    y_lower += len(ith_silhouette_vals)
plt.xlabel('Silhouette Score')
plt.ylabel('Sample Index')
plt.title('Diagram 2. Visualization of Silhouette Score in Plantation Asset Clustering Results')
plt.show()
```

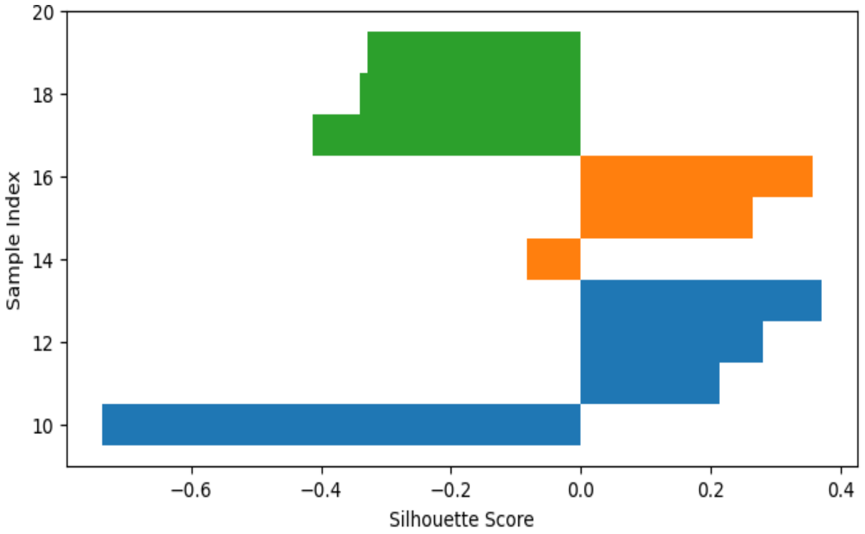


Diagram 2. Visualization of Silhouette Score in Plantation Asset Clustering Results

V. APPENDIX: CODE AND TECHNICAL IMPLEMENTATION

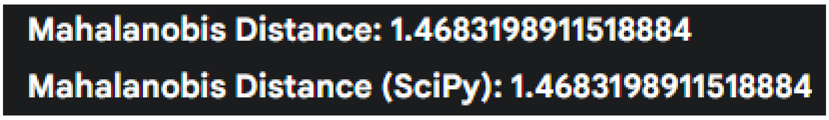
5.1. Mahalanobis Distance Calculation Code Implementation

This section will briefly explain the significance of Mahalanobis distance in plantation asset clustering.

Appendix C.

```
import numpy as np
from scipy.spatial import distance

#Example of asset vector data and mean
        x = np.array([2, 3, 4])
        mu = np.array([1, 2, 3])

#Covariance matrix of asset data
        S = np.array([[1, 0.2, 0.1],
                [0.2, 1, 0.3],
                [0.1, 0.3, 1]])

# Calculating Mahalanobis distance manually
        diff = x - mu
        S_inv = np.linalg.inv(S)
        mahalanobis_distance = np.sqrt(np.dot(np.dot(diff.T, S_inv), diff))
        print("Mahalanobis Distance:", mahalanobis_distance)

#Alternative using SciPy's Built-in Function
        mahalanobis_distance_scipy = distance.mahalanobis(x, mu, S_inv)
        print("Mahalanobis Distance (SciPy):", mahalanobis_distance_scipy)
```

Program Output:



```
Mahalanobis Distance: 1.4683198911518884
Mahalanobis Distance (SciPy): 1.4683198911518884
```

Diagram 3: Output of Mahalanobis Distance Calculation Using Python

5.2. Implementation of Hierarchical Clustering Code

Define a function that performs hierarchical clustering on plantation asset data, where the previously calculated Mahalanobis distance matrix is used as the basis for grouping. This process utilizes Python libraries such as SciPy and scikit-learn to build the cluster structure and generate a dendrogram visualization. Linkage parameters and the Mahalanobis distance method are used to determine the clustering results, and this code implementation can be saved as a Python file for further analysis.

Appendix D

```
import numpy as np
from scipy.cluster.hierarchy import linkage, dendrogram
import matplotlib.pyplot as plt
from scipy.spatial.distance import pdist, squareform

# Sample asset data (each row represents an asset, each column is a feature)
data = np.array([
    [2, 3, 4],
    [1, 2, 3],
    [2, 2, 2],
    [4, 5, 6]
])

# Calculate the Mahalanobis distance matrix.
VI = np.linalg.inv(np.cov(data, rowvar=False))
mahal_dist = pdist(data, metric='mahalanobis', VI=VI)
# Perform hierarchical clustering using the linkage method
Z = linkage(mahal_dist, method='ward')  # The 'ward' method is often used for balanced cluster results.

# Dendrogram visualization
plt.figure(figsize=(8, 4))
dendrogram(Z, labels=['Asset1', 'Asset2', 'Asset3', 'Asset4'])
plt.title(Dendrogram of Hierarchical Clustering Results)
plt.xlabel('Asset')
plt.ylabel('Distance (Mahalanobis)')
plt.show()
```

Now, our goal is to hierarchically cluster plantation assets using the calculated Mahalanobis distance matrix. The Mahalanobis distance can be computed manually by first calculating the difference between individual data vectors and their mean vector. This difference is then used in a formula involving the square root of the product of the transpose of that difference, the inverse of the covariance matrix, and the data difference itself. Alternatively, SciPy's built-in functions can be utilized, for instance, by calling the distance.mahalanobis function, which accepts individual data, the mean, and the inverse covariance matrix as inputs. The distances obtained will then be used with SciPy's linkage and dendrogram functions in Python for the hierarchical clustering process. The resulting dendrogram visualization will facilitate interpreting the relationships between assets during the clustering process.

5.3 Code Integration for Big Data Analytics and Algorithm Modeling

Big data analytics enables fast and effective processing of plantation asset data through preprocessing stages such as data cleaning, normalization, and handling of missing values. This process is crucial for ensuring data quality before further analysis. For large-scale data, the ETL (Extract, Transform, Load) process is used to extract data from various sources, transform it to meet analysis needs, and load it into an efficiently accessible storage system[25]. The implementation of ETL processes and big data analytics can be carried out using libraries such as PySpark or Dask, which support parallel and distributed data processing. This code example demonstrates how asset data can be processed using PySpark to perform normalization and handle missing values.

After the data undergoes careful preprocessing and yields a set of scaled features, as intuitively visualized in Diagram 4, clustering algorithms like K-Means can be applied to identify patterns and groups within the asset data. Diagram 4, which we refer to as the Scaled Feature Data Visualization (3D Scatter Plot), effectively presents a representation of plantation asset data in a three-dimensional space, where each point reflects one asset with three scaled features as its X, Y, and Z coordinates. This initial visualization provides an exploratory overview of the data distribution, helping to identify potential clusters before formal modeling is performed.

The mathematical core of the K-Means algorithm applied is the minimization of the objective function or inertia. This formula measures how compact the formed clusters are, aiming to minimize the total squared distance from each data point to its nearest cluster centroid. In summary, the mathematical formula for inertia is:

$$J = \sum_{j=1}^{K} \sum_{x_i \epsilon S_j} \left\| x_i - \mu_j \right\|^2 \rightarrow 9[26]$$

Where: J: The inertia value to be minimized; K: The desired number of clusters; Sj: The set of data points in the j-th cluster; xi: The i-th individual data point; μj: The centroid of the j-th cluster; and ||·||2: The squared Euclidean distance.

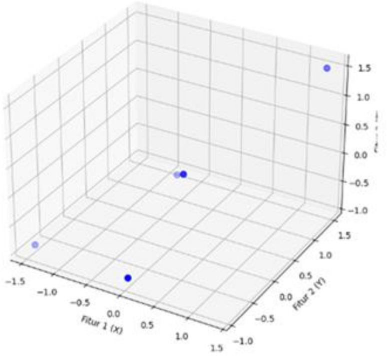


Diagram 4: Visualisasi Data Fitur Berskala (3D Scatter Plot)

5.4 Data Security Implementation with Blockchain

This section outlines the practical implementation and fundamental mechanisms for securing clustering results data using blockchain technology. Just as Lyapunov exponents explain chaotic behavior, this section will describe how blockchain deterministically ensures data integrity and auditability within complex systems.

Data security implementation begins with storing clustering results into a blockchain system. This process involves writing code or pseudocode responsible for converting analysis results into digital transactions that can be recorded, ensuring each cluster or related asset risk information is immutably entered into a block. Furthermore, a simple smart contract code example (e.g., in Solidity for Ethereum or Hyperledger Fabric) can be presented to illustrate how business logic and validation rules can be automated. These smart contracts play a crucial role in verifying the validity of clustering results data before it is added to the chain, as well as managing access control for authorized parties. Fundamentally, the recording mechanism in blockchain ensures that every stored data point has a cryptographic hash trace linked to the previous block, forming an unmanipulable chain. This significantly enhances auditability, allowing historical verification of all changes and analyses performed on plantation asset data. Thus, the security of clustering results data relies not only on encryption but also on the decentralized and consensus-based nature of blockchain, which is resistant to unauthorized alterations, ensuring transparency and trust among stakeholders. The output of this recording mechanism can be traced as transaction trails, as shown in Diagram 5.

.

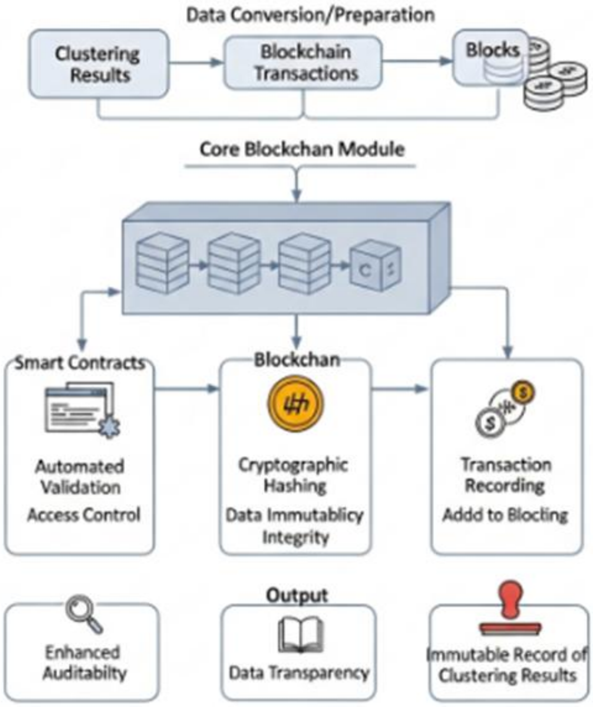
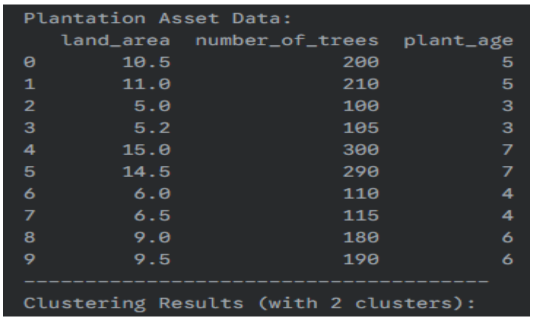


Diagram 5: Data Security Architecture for Clustering Results Using Blockchain

5.5 Evaluation and Visualization of Results (Optional)

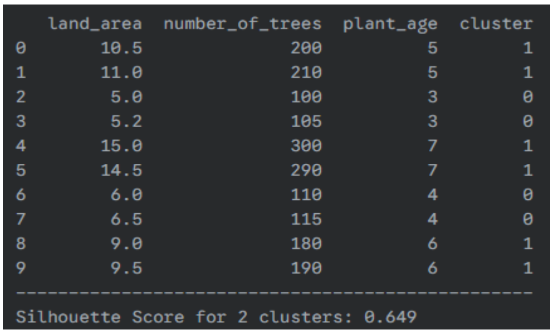
This section discusses the process of quantitative evaluation and visualization of clustering results to gain a deeper qualitative understanding of the developed system's behavior. To measure the effectiveness and quality of clustering, program code can be used to calculate evaluation metrics such as the Silhouette Score. This calculation is crucial for validating cluster homogeneity and inter-cluster separation, similar to how metrics in physical systems evaluate behavioral characteristics.

Although it is difficult to precisely measure system dynamics solely from such plots, visualizations of clustering results (e.g., scatter plots) can be generated using code. These plots provide a qualitative overview of how assets are grouped and indicate certain behavioral patterns within the system, similar to how variable plots in physical systems can show periodic behavior. With the example interpretations of the visualization results presented, it is possible to review the simulations and check whether the underlying theory aligns with the observed behavior. Graphs related to clustering result visualization and other evaluations are presented in Diagram 6.



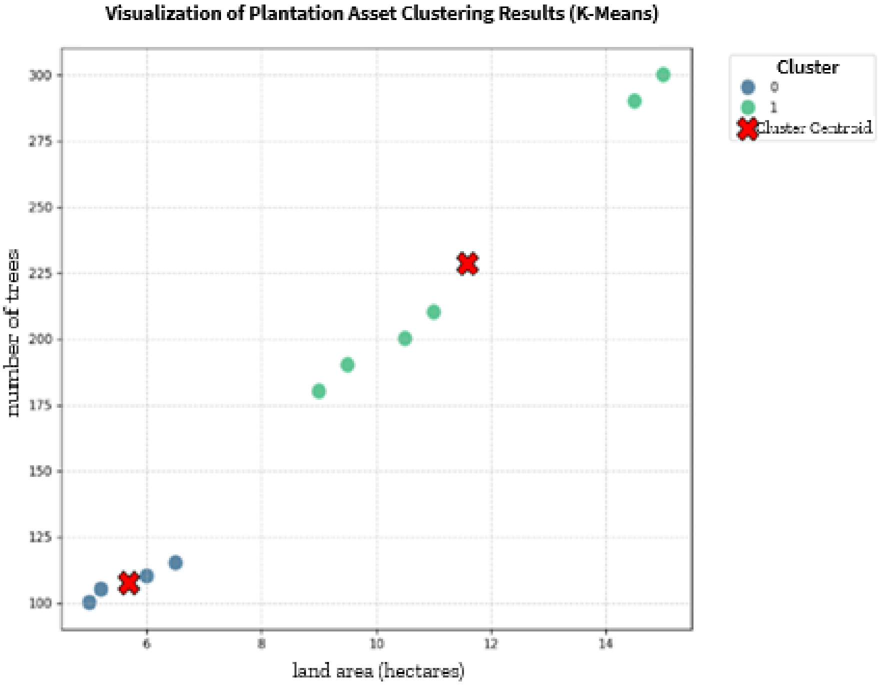
```
Plantation Asset Data:
   land_area  number_of_trees  plant_age
0       10.5              200          5
1       11.0              210          5
2        5.0              100          3
3        5.2              105          3
4       15.0              300          7
5       14.5              290          7
6        6.0              110          4
7        6.5              115          4
8        9.0              180          6
9        9.5              190          6
----------------------------------------
Clustering Results (with 2 clusters):
```

(a)

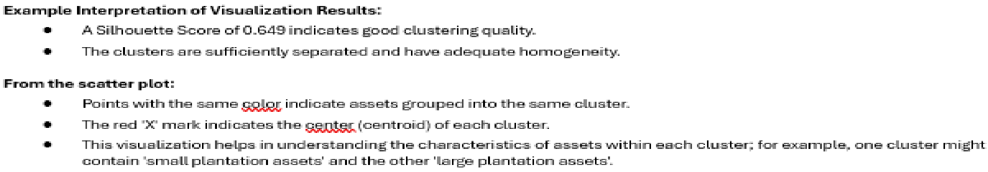


```
   land_area  number_of_trees  plant_age  cluster
0       10.5              200          5        1
1       11.0              210          5        1
2        5.0              100          3        0
3        5.2              105          3        0
4       15.0              300          7        1
5       14.5              290          7        1
6        6.0              110          4        0
7        6.5              115          4        0
8        9.0              180          6        1
9        9.5              190          6        1
--------------------------------------------------
Silhouette Score for 2 clusters: 0.649
```

(b)



(c)



Example Interpretation of Visualization Results:

- A Silhouette Score of 0.649 indicates good clustering quality.
- The clusters are sufficiently separated and have adequate homogeneity.

From the scatter plot:

- Points with the same color indicate assets grouped into the same cluster.
- The red 'X' mark indicates the center (centroid) of each cluster.
- This visualization helps in understanding the characteristics of assets within each cluster; for example, one cluster might contain 'small plantation assets' and the other 'large plantation assets'.

(d)

Diagram 6: Visualization of Plantation Asset Clustering Results (K-Means)

VI. PERFORMANCE ANALYSIS AND SYSTEM LIMITATIONS

In this chapter, with the plantation asset cybersecurity framework already modeled and algorithmically formulated, we will now investigate its performance. This chapter will discuss the evaluation of the framework's performance under various operational conditions, as well as identify the inherent limitations and complexities of the developed model. Similar to how the linearization process analyzes the behavior of physical systems and their model limitations, here we will explain how this framework interacts with real-world data and challenges. We will see how this framework operates under varying data scenarios, measure its effectiveness, and understand where its complexities lie and where there is room for further development. Let's explain how we analyze the performance and limitations of this framework through the following steps:

Step (1) We must consider Framework Performance Evaluation under Varying Conditions. This involves testing the framework under diverse data and operational scenarios to measure its response and effectiveness, similar to defining the mechanical simulation environment and physical parameters in a pendulum model.

Step (2) Identification of Model Limitations and Complexities. At this stage, we will carefully identify the inherent limitations of the developed model, such as dependence on clustering parameters or the operational complexity of blockchain, similar to how a revolute joint represents degrees of freedom and their limitations.

Step (3) Discussion of Potential System Improvements and Adaptations. Once limitations are identified, we will propose strategies to overcome these challenges, such as advanced optimization or real-time implementation, analogous to how sensors measure and provide data for system adjustments.

Then, as this analysis is conducted, we will obtain various different results and findings, namely the analysis of framework performance under varying conditions, the identification of model limitations and complexities, and the potential for system improvements and adaptations, each of which will be represented in detail in the subsequent sub-chapters.

6.1 Framework Performance Analysis under Varying Conditions

Considering the framework's performance under varying data conditions, evaluating it with relevant metrics, and presenting the interpreted results, as presented by Diagram 7.

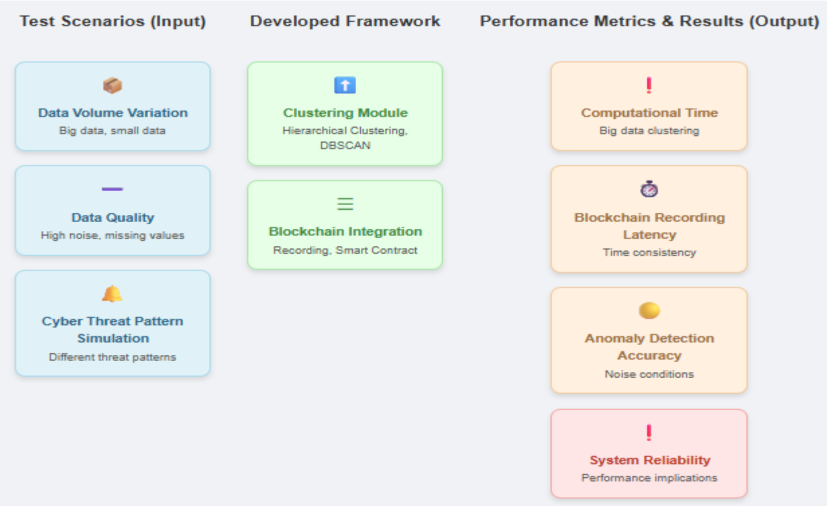


Diagram 7: Framework Performance Analysis under Varying Conditions

6.2 Model Limitations and Complexities

In this section, we will identify the inherent limitations and complexities of the developed framework. Similar to how a linearization model of a physical system is a simplification that has its own boundaries from initial conditions, this plantation asset cybersecurity framework also faces challenges inherent in its design and implementation.

Parameter Dependency: The effectiveness of clustering within this framework heavily relies on the appropriate selection of parameters, such as the desired number of clusters, the distance metric used (e.g., Mahalanobis Distance), and algorithm-specific parameters like epsilon (ϵ) and MinPts for DBSCAN. Determining optimal parameters often requires extensive experimentation and deep domain understanding, which can be a significant challenge in the context of dynamic and varied plantation data.

Operational Complexity of Blockchain: The integration of blockchain into the framework, while providing security and auditability guarantees, also adds a substantial layer of operational complexity. This includes challenges related to blockchain network maintenance, managing consensus among nodes, and system scalability to handle ever-increasing data volumes in real-time. The process of recording transactions and verifying smart contracts requires computational resources and careful management to ensure efficiency without sacrificing data security.

Multidimensional and Dynamic Data Challenges: This framework is designed to handle very large and multidimensional data from various sources such as IoT, GIS, and financial data. However, the dynamic nature of this data, including changing patterns of cyber threats and variations in plantation operational conditions, remains a challenge. Ensuring that the clustering models and blockchain security mechanisms can effectively adapt to these changes without requiring frequent manual reconfiguration is an aspect of complexity that needs continuous exploration.

Computational Complexity of Clustering Algorithms: Computational complexity is an important limitation, especially when dealing with big data. For clustering algorithms like K-Means, the time complexity can generally be represented as:

$$O(I \cdot K \cdot N \cdot D)$$

Where: **I** is the number of iterations required for algorithm convergence; **K** is the desired number of clusters; **N** is the number of data points (plantation assets); and **D** is the number of dimensions or data features. This formula indicates that the computational time will increase linearly with the number of iterations, the number of clusters, the number of data points, and the number of features, highlighting the framework's scalability challenges when applied to very large and high-dimensional datasets.

6.3 Potential System Improvements and Adaptations

This section discusses future development directions or how identified limitations can be addressed. Similar to how linearization error analysis informs model improvements, this framework has the potential to be enhanced to overcome existing complexities. This involves advanced optimization of clustering parameters and exploration of alternative algorithms, as well as real-time implementation for data streaming scenarios. Improvements in blockchain scalability and efficiency are also considered through the exploration of hybrid blockchain or other distributed ledger technologies. Finally, the development of adaptive threat detection is proposed through the integration of the framework with advanced models such as Generative Adversarial Networks (GANs). The potential for system improvements and adaptations is visualized in Diagram 8.

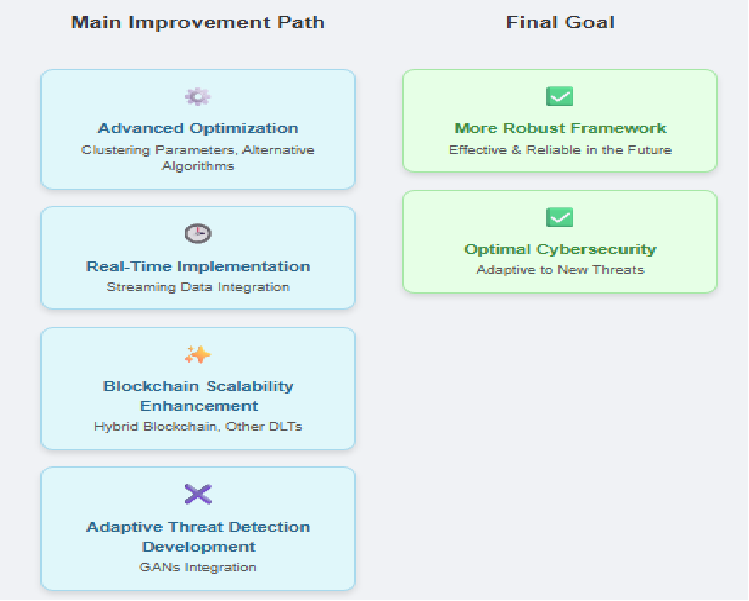

Main Improvement Path
Final Goal
Advanced Optimization
Clustering Parameters, Alternative Algorithms
More Robust Framework
Effective & Reliable in the Future
Real-Time Implementation
Streaming Data Integration
Optimal Cybersecurity
Adaptive to New Threats
Blockchain Scalability Enhancement
Hybrid Blockchain, Other DLTs
Adaptive Threat Detection Development
GANs Integration

Diagram 8. Potential Improvements and Adaptations of the Cybersecurity Framework System

Diagram 8 illustrates various development paths that can be taken to strengthen the plantation asset cybersecurity framework, ensuring its ability to adapt and perform optimally in the future.

VII. CONCLUSION

The plantation asset cybersecurity system that integrates Big Data, Hierarchical Clustering, and Blockchain is a highly complex system. Given the complexity of multidimensional data, the dynamic nature of cyber threats, and the operational challenges of distributed technologies, there are numerous assumptions and conditions that influence the framework's performance. The inability to adapt to changing data conditions or cyber-attacks can lead to a state resembling chaos, namely a condition of significant disorder and uncertainty in risk management. Chaos in this context, meaning uncertainty over time, is highly sensitive to the initial conditions of the data and system parameters, and can only occur if the system is not managed conservatively. However, response time and threat detection effectiveness, analogous to the period of motion, do not depend on the volume of raw data or the scale of assets being analyzed. Other factors involved in the success of this system are the accuracy of the clustering algorithms and the guaranteed integrity of the blockchain.

REFERENCES

[1] S. M. Frank, M. R. Maechler, S. V. Fogelson, and P. U. Tse, “Hierarchical categorization learning is associated with representational changes in the dorsal striatum and posterior frontal and parietal cortex,” *Hum Brain Mapp*, vol. 44, no. 9, pp. 3897–3912, Jun. 2023, doi: 10.1002/hbm.26323.

[2] C. Atik, “Towards Comprehensive European Agricultural Data Governance: Moving Beyond the ‘Data Ownership’ Debate,” *IIC International Review of Intellectual Property and Competition Law*, vol. 53, no. 5, pp. 701–742, May 2022, doi: 10.1007/s40319-022-01191-w.

[3] S. He, X. Xing, G. Wang, and Z. Sun, “A Data Integrity Verification Scheme for Centralized Database Using Smart Contract and Game Theory,” *IEEE Access*, vol. 11, pp. 59675–59687, 2023, doi: 10.1109/ACCESS.2023.3284850.

[4] R. Vatankhah Barenji, “A blockchain technology based trust system for cloud manufacturing,” *J Intell Manuf*, vol. 33, no. 5, pp. 1451–1465, Jun. 2022, doi: 10.1007/s10845-020-01735-2.

[5] Z. Sun, D. Han, D. Li, X. Wang, C. C. Chang, and Z. Wu, “A blockchain-based secure storage scheme for medical information,” *EURASIP J Wirel Commun Netw*, vol. 2022, no. 1, Dec. 2022, doi: 10.1186/s13638-022-02122-6.

[6] F. Kuntke, S. Linsner, E. Steinbrink, J. Franken, and C. Reuter, “Resilience in Agriculture: Communication and Energy Infrastructure Dependencies of German Farmers,” *International Journal of Disaster Risk Science*, vol. 13, no. 2, pp. 214–229, Apr. 2022, doi: 10.1007/s13753-022-00404-7.

[7] K. Martens and J. Zscheischler, “The Digital Transformation of the Agricultural Value Chain: Discourses on Opportunities, Challenges and Controversial Perspectives on Governance Approaches,” *Sustainability (Switzerland)*, vol. 14, no. 7, Apr. 2022, doi: 10.3390/su14073905.

[8] A. R. Abdulai, K. B. KC, and E. Fraser, “What factors influence the likelihood of rural farmer participation in digital agricultural services? experience from smallholder digitalization in Northern Ghana,” *Outlook Agric*, vol. 52, no. 1, pp. 57–66, Mar. 2023, doi: 10.1177/00307270221144641.

[9] J. A. Stenberg, F. Nakazi, and H. Sekabira HSekabira, “Are digital services the right solution for empowering smallholder farmers? A perspective enlightened by COVID-19 experiences to inform smart IPM.”

[10] F. Zhou, X. Du, W. Li, Z. Lu, and J. Wu, “NIDD: an intelligent network intrusion detection model for nursing homes,” *Journal of Cloud Computing*, vol. 11, no. 1, Dec. 2022, doi: 10.1186/s13677-022-00361-y.

[11] L. Y. Q. Chong and T. S. Lim, “Pull and Push Factors of Data Analytics Adoption and Its Mediating Role on Operational Performance,” *Sustainability (Switzerland)*, vol. 14, no. 12, Jun. 2022, doi: 10.3390/su14127316.

[12] Á. Regan, “Exploring the readiness of publicly funded researchers to practice responsible research and innovation in digital agriculture,” *J Responsible Innov*, vol. 8, no. 1, pp. 28–47, 2021, doi: 10.1080/23299460.2021.1904755.

[13] M. Vichi, C. Cavicchia, and P. J. F. Groenen, “Hierarchical Means Clustering,” *J Classif*, vol. 39, no. 3, pp. 553–577, Nov. 2022, doi: 10.1007/s00357-022-09419-7.

[14] L. Yu, L. Yu, and K. Yu, “A high-dimensionality-trait-driven learning paradigm for high dimensional credit classification,” *Financial Innovation*, vol. 7, no. 1, Dec. 2021, doi: 10.1186/s40854-021-00249-x.

[15] T. Li, G. Kou, Y. Peng, and P. S. Yu, “An Integrated Cluster Detection, Optimization, and Interpretation Approach for Financial Data,” *IEEE Trans Cybern*, vol. 52, no. 12, pp. 13848–13861, Dec. 2022, doi: 10.1109/TCYB.2021.3109066.

[16] L. Yang, S. X. Yang, Y. Li, Y. Lu, and T. Guo, “Generative Adversarial Learning for Trusted and Secure Clustering in Industrial Wireless Sensor Networks,” *IEEE Transactions on Industrial Electronics*, vol. 70, no. 8, pp. 8377–8387, Aug. 2023, doi: 10.1109/TIE.2022.3212378.

[17] Y. Javed, M. A. Khayat, A. A. Elghariani, and A. Ghafoor, “PRISM: A Hierarchical Intrusion Detection Architecture for Large-Scale Cyber Networks,” Nov. 2021, doi: 10.1109/TDSC.2023.3240315.

[18] Z. Yang *et al.*, “DC-FUDA: Improving deep clustering via fully unsupervised domain adaptation,” *Neurocomputing*, vol. 526, pp. 109–120, Mar. 2023, doi: 10.1016/j.neucom.2023.01.058.

[19] M. Landauer, F. Skopik, M. Wurzenberger, and A. Rauber, “System log clustering approaches for cyber security applications: A survey,” *Comput Secur*, vol. 92, May 2020, doi: 10.1016/j.cose.2020.101739.

[20] W. Mu, K. H. Lim, J. Liu, S. Karunasekera, L. Falzon, and A. Harwood, “A clustering-based topic model using word networks and word embeddings,” *J Big Data*, vol. 9, no. 1, Dec. 2022, doi: 10.1186/s40537-022-00585-4.

[21] B. Corgnet, C. Deck, M. DeSantis, K. Hampton, and E. O. Kimbrough, “When Do Security Markets Aggregate Dispersed Information?,” *Manage Sci*, vol. 69, no. 6, pp. 3697–3729, Jun. 2023, doi: 10.1287/mnsc.2022.4463.

[22] M. Zhou *et al.*, “Transformation-Based Fuzzy Rule Interpolation With Mahalanobis Distance Measures Supported by Choquet Integral,” *IEEE Transactions on Fuzzy Systems*, vol. 31, no. 4, pp. 1083–1097, Apr. 2023, doi: 10.1109/TFUZZ.2022.3194368.

[23] J. Anibal *et al.*, “HAL-X: Scalable hierarchical clustering for rapid and tunable single-cell analysis,” *PLoS Comput Biol*, vol. 18, no. 10, Oct. 2022, doi: 10.1371/journal.pcbi.1010349.

[24] H. Chai, S. Leng, J. He, K. Zhang, and B. Cheng, “CyberChain: Cybertwin Empowered Blockchain for Lightweight and Privacy-Preserving Authentication in Internet of Vehicles,” *IEEE Trans Veh Technol*, vol. 71, no. 5, pp. 4620–4631, May 2022, doi: 10.1109/TVT.2021.3132961.

[25] M. Pedrera-Jiménez *et al.*, “TransformEHRs: A flexible methodology for building transparent ETL processes for EHR reuse,” *Methods Inf Med*, vol. 61, no. 5, pp. E89–E102, Dec. 2022, doi: 10.1055/s-0042-1757763.

[26] W. Qian, Y. Zhang, and Y. Chen, “Structures of Spurious Local Minima in k-means,” Feb. 2020, [Online]. Available: http://arxiv.org/abs/2002.06694