International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

User and Entity Behavior Analytics for Cybersecurity Using Unsupervised Clustering Techniques

Dr.K.Subbarao[1], Ambika Jyothi Devana[2], Nandini Meda[3], Madhu Kumar Nidrabingi[4], Rohith Sai Pasupuleti[5]
Professor & HOD[1], Student[2,3,4,5] Department of CSE – Data Science
St. Ann's College of Engineering & Technology, Chirala, Andhra Pradesh, India.

Abstract — With the increasing complexity of cyber threats, especially insider attacks and unknown anomalies, traditional rule-based security systems are often insufficient. This paper presents a User and Entity Behavior Analytics (UEBA) system that applies unsupervised machine learning techniques to analyze user and system activity logs and detect abnormal behavior. Since labeled attack data is rarely available in real-world scenarios, clustering algorithms are employed to learn normal behavior patterns and identify deviations automatically. Multiple clustering algorithms — K-Means, DBSCAN, and HDBSCAN — are implemented and evaluated on the E-shop Clothing Clickstream Dataset comprising over 165,000 records. Data preprocessing encompasses label encoding of categorical attributes and Z-score normalization to ensure uniform feature contribution to clustering. Performance is assessed using internal evaluation metrics: Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Index. Experimental results demonstrate that density-based algorithms, particularly HDBSCAN, outperform partition-based methods in identifying irregular and rare behavior patterns. HDBSCAN automatically identifies anomalous sessions as noise points (Cluster -1) without requiring a predefined number of clusters. The system is deployed as an interactive Streamlit web application, providing security analysts with cluster visualization, anomaly export, and behavioral profiling capabilities. The proposed approach demonstrates a scalable and practical solution for proactive cybersecurity monitoring applicable across banking, e-commerce, healthcare, and enterprise security domains.

Keywords — UEBA; anomaly detection; HDBSCAN; unsupervised clustering; cybersecurity; clickstream analytics; insider threat; behavioral analytics

I. INTRODUCTION

A. Introduction to UEBA

User and Entity Behavior Analytics (UEBA) is a cybersecurity process that involves analyzing the behavior of users and entities — such as devices, applications, and servers — to detect anomalies that may indicate security threats. Unlike traditional security systems that rely on signature-based detection and rule engines, UEBA builds behavioral baselines from historical data and flags deviations in real time. The concept of UEBA evolved from earlier User Behavior Analytics (UBA) solutions and now extends monitoring to all entities within a network, combining data science, machine learning, and cybersecurity principles to create adaptive, intelligent monitoring systems [1].

B. Overview of Cybersecurity and Behavior Analytics

The cybersecurity landscape has undergone dramatic transformation over the past two decades. Early security measures focused on perimeter defense using firewalls and antivirus tools. As networks became increasingly complex, the industry evolved toward intrusion detection systems (IDS), security information and event management (SIEM), and ultimately behavior analytics. Table I summarizes this evolution across different eras, highlighting how defensive technologies have progressed alongside emerging threats.

TABLE I. Evolution of Cybersecurity Threat Landscape

Era	Primary Threat	Dominant Defense	Limitation
1990s	Viruses, Worms	Antivirus, Firewalls	Signature-dependent; unknown threats evade detection
2000s	Network Intrusions, DDoS	IDS/IPS, SIEM	High false positive rates; cannot detect insider threats
2010s	APTs, Ransomware, Insider Threats	IDS/IPS, SIEM	Lack of behavioral context; requires labeled threat data
2020s	AI-Driven Attacks, Zero-Days	UEBA, Zero Trust Architecture	Requires continuous tuning; complex deployment

Behavior analytics represents the fourth generation of cybersecurity defense, combining the best elements of all previous approaches while adding adaptive, data-driven intelligence. Modern UEBA platforms integrate with SIEM and SOAR (Security Orchestration, Automation and Response) platforms to provide comprehensive organizational protection.

C. Rule-Based vs. Behavior-Based Security

Traditional rule-based security systems operate on predefined signatures and thresholds. While effective against known threats, they fail to adapt to novel attack patterns, zero-day vulnerabilities, and sophisticated insider threats. Table II provides a comparative analysis illustrating how behavior-based security overcomes the fundamental limitations of rule-based approaches.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

TABLE II. Rule-Based vs. Behavior-Based Security Comparison

Criterion	Rule-Based Security	Behavior-Based Security (UEBA)
Detection Basis	Predefined signatures and static rules	Dynamic behavioral baselines learned from data
Adaptability	Cannot detect new/unknown threats	Automatically adapts to evolving threat patterns
Insider Threat Detection	Limited; rules rarely cover authorized users behaving maliciously	Highly effective; detects subtle deviations in authorized user behavior
False Positive Rate	High; any policy violation triggers an alert	Lower; anomalies must deviate significantly from baseline
Labeled Data Requirement	Requires known threat signatures	No labeled data required (unsupervised approach)
Scalability	Rule sets become unwieldy at scale	Models scale with data volume and dimensionality

D. Problem Statement

Modern organizations generate massive volumes of security logs from servers, applications, network devices, and user workstations. The sheer volume of data makes manual analysis impractical — a typical enterprise may generate millions of log entries per day. The core challenges addressed in this work are: (1) insider threats and anomalous behaviors cannot be detected by rule-based systems that require predefined attack signatures; (2) existing UEBA solutions frequently rely on one or two clustering techniques without systematic comparative evaluation; (3) selecting the most appropriate clustering algorithm for different UEBA scenarios with varying data density and dimensionality remains an open research challenge [3].

E. Objectives of the Project

The primary objective of this work is to apply and evaluate clustering-based unsupervised learning techniques for UEBA. Specific objectives include: implementing and comparing K-Means, DBSCAN, and HDBSCAN clustering algorithms for behavior-based anomaly detection; preprocessing the E-shop Clothing Clickstream Dataset using label encoding and Z-score normalization; evaluating clustering performance using Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Index; and demonstrating the effectiveness of HDBSCAN for practical UEBA applications deployable in enterprise environments.

II. LITERATURE SURVEY

A. Existing UEBA and Anomaly Detection Approaches

UEBA and behavior-based anomaly detection have been studied from multiple angles: statistical modeling, machine learning, deep learning, and hybrid approaches. Table III summarizes key existing systems reviewed in this study.

TABLE III. Existing UEBA and Anomaly Detection Approaches

Title / Authors	Methodology	Advantages	Disadvantages
Shashanka et al. – User Behavior Analytics for Insider Threat Detection	ML-based behavioral analysis with SVD dimensionality reduction	Detects anomalies without predefined signatures; reduces data complexity	Limited interpretability; sensitive to data quality issues
Tang et al. – Feature Engineering for Behavior-Based Anomaly Detection	Feature engineering with behavior-based anomaly detection models	Improves detection accuracy; extracts meaningful behavioral features	Requires manual feature selection; limited generalization to new domains
Singh et al. – Clustering-Based Intrusion Detection in Network Traffic	K-Means and DBSCAN applied to network flow data	No labeled data required; detects novel attacks	Difficulty setting parameters; unstable across different traffic profiles

B. Types of Inputs Used in UEBA Models

UEBA systems integrate multiple data sources to construct comprehensive behavioral profiles. Table IV categorizes the primary data types used in modern UEBA systems and their relevance to anomaly detection.

TABLE IV. Types of Inputs Used in UEBA Models

Data Category	Specific Features	Source	Relevance to UEBA
Behavioral / Clickstream	Page views, session duration, click sequences, navigation paths	Web server logs, application logs	Captures user intent and browsing patterns; reveals unusual navigation
Technical Indicators	Login times, IP addresses, device fingerprints, access frequency	Authentication logs, AD/LDAP	Identifies suspicious access patterns and location anomalies
Network Activity	Connection volume, port access, data transfer size, protocol usage	Network flow data, firewall logs	Detects data exfiltration and lateral movement

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Entity Metadata	Country, product category, user role, department	HR systems, directory services	Contextualizes behavior against peer group norms
Temporal Features	Time of access, day of week, session gap patterns	Derived from timestamps	Identifies off-hours access and irregular session patterns

C. Feature Selection Methods

Feature selection is critical for improving model accuracy and efficiency. In high-dimensional behavioral datasets, many features may be redundant or correlated, leading to overfitting and increased computational cost. Table V summarizes key feature selection techniques applied in this study.

TABLE V. Feature Selection Methods

Method	Description	Use Case in UEBA	Pros	Cons
Correlation Analysis	Removes highly correlated features to reduce redundancy	Eliminating redundant behavioral metrics	Simple; interpretable	Only detects linear relationships
PCA	Transforms correlated features into uncorrelated principal components	Dimensionality reduction for visualization and clustering	Reduces noise; improves clustering	Components lose original interpretability
Label Encoding	Converts categorical features to numerical format	Encoding country, product group, color attributes	Lightweight; compatible with all algorithms	Implies ordinal relationships in nominal data

D. Related Work

Shashanka et al. introduced machine learning-based behavioral analytics for insider threat detection, applying Singular Value Decomposition (SVD) for dimensionality reduction. While effective, their approach lacked interpretability. Tang et al. proposed feature engineering pipelines demonstrating that carefully constructed behavioral features significantly improve detection accuracy. Singh et al. applied Long Short-Term Memory (LSTM) networks for sequential user behavior modeling, achieving high detection accuracy but requiring labeled training data and substantial computational resources [3]. Campello et al. [2] introduced HDBSCAN as an extension of DBSCAN based on hierarchical density estimates, providing the theoretical foundation for the primary algorithm adopted in this work. Chandola et al. [3] published a comprehensive survey on anomaly detection techniques, offering the conceptual taxonomy within which this study's approach is situated. MacQueen et al. introduced the K-Means clustering algorithm used for baseline comparison in this project [7].

III. PROPOSED METHODOLOGY

A. Dataset Description

The proposed system utilizes the E-shop Clothing Clickstream Dataset to model user-entity interactions within an online shopping environment. This dataset captures user browsing activities and interaction patterns on an e-commerce platform, including product views, session behaviors, and navigation activities. It comprises 165,473 records with 14 features spanning three primary attribute categories, as summarized in Table VI.

TABLE VI. Dataset Characteristics

Attribute Type	Description	Purpose
Browsing Data	Page visits, navigation paths	Understand user behavior
Product Data	Product category, type	Analyze product interactions
Session Data	Duration, actions, entry/exit pages	Session-level analysis

B. Data Preprocessing

Data preprocessing transforms raw clickstream data into a structured format suitable for machine learning. This stage encompasses three primary steps. First, attribute separation organizes mixed browsing, product, and session attributes into structured columns to improve clarity and processing efficiency. The page2 feature is parsed into two components: model_group (categorical: A, B, C, P) and model_id (numeric: 1-82). The constant year column is removed as it contributes no discriminative information.

Second, Label Encoding is applied to convert categorical attributes — including product categories, page types, and country identifiers — into numerical representations, enabling machine learning algorithms to process categorical data effectively. Third, Z-score Normalization is performed using StandardScaler, ensuring all features contribute equally to the clustering process regardless of their original measurement scale.

C. Feature Engineering

Meaningful behavioral features are extracted from the raw dataset and organized into three categories as described in Table VII. These features enhance clustering performance and support effective identification of abnormal activities in UEBA systems.

TABLE VII. Key Features Used

Feature Category	Features	Description
Browsing Behavior	Pages visited, click frequency, navigation paths	Captures user browsing patterns

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Product Interaction	Product category, type, interaction frequency	Tracks user-product engagement
Session-Based	Session duration, actions, entry/exit pages	Represents session dynamics

D. Unsupervised Clustering Algorithms

Three unsupervised clustering algorithms are implemented and compared. K-Means partitions data into k clusters by minimizing intra-cluster variance. It is computationally efficient but requires specifying k in advance and assumes spherical, equally-sized clusters. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) groups points in dense regions and marks sparse outliers as noise, automatically detecting anomalies without requiring a cluster count. However, it requires careful tuning of the epsilon and min_samples hyperparameters [1]. HDBSCAN (Hierarchical DBSCAN) extends DBSCAN by building a cluster hierarchy and automatically selecting stable clusters using the excess-of-mass (EOM) method, handling variable-density data and identifying noise points — interpreted as anomalies in the UEBA context — without requiring a predefined cluster count [2][4].

E. Cluster Evaluation Metrics

Internal validation metrics are employed to assess clustering quality without requiring ground-truth labels, as summarized in Table VIII.

TABLE VIII. Evaluation Metrics

Metric	Description	Ideal Value
Silhouette Score	Measures similarity within clusters; ranges from -1 to 1	Higher is better
Calinski-Harabasz Index	Ratio of between-cluster to within-cluster variance	Higher is better
Davies-Bouldin Index	Measures average similarity between each cluster and its most similar cluster	Lower is better

The Silhouette Score quantifies cluster cohesion and separation by comparing, for each data point, the mean intra-cluster distance (a) against the mean nearest-cluster distance (b). The score equals (b - a) / max(a, b), with values closer to +1 indicating well-separated clusters [7]. The Davies-Bouldin Index measures average similarity between each cluster and its most similar cluster, with lower values indicating better-defined partitions [8]. The Calinski-Harabasz Index evaluates the ratio of between-cluster dispersion to within-cluster dispersion, with higher values representing well-defined, compact clusters [9].

F. Behavioral Analysis for UEBA

Following clustering, the resulting groups are analyzed to understand user behavior patterns. Clusters represent either normal browsing behavior or anomalous activity. Behavioral analysis focuses on continuously monitoring user activities and comparing them against established baseline patterns derived from clustering results. The system examines factors such as session duration, page transition frequency, frequency of product interactions, and access timing to distinguish typical from atypical behaviors. This enables early detection of insider threats and compromised accounts while reducing false positives compared to traditional rule-based approaches.

IV. SYSTEM DESIGN

A. System Workflow

The UEBA system processes data through a sequential pipeline of five stages. Stage 1 (Data Collection): Raw clickstream data is ingested from the E-shop Clothing Dataset, capturing page visits, click events, session duration, navigation paths, and interaction frequency. Stage 2 (Data Preprocessing): Raw data is cleaned by removing duplicates and missing values, categorical attributes are encoded using Label Encoding, and feature values are standardized using Z-score Normalization. Stage 3 (Feature Engineering): Meaningful behavioral patterns are extracted, including number of clicks per session, page visit frequency, time spent on pages, and navigation sequence patterns. Stage 4 (HDBSCAN Clustering): The HDBSCAN algorithm is applied to the normalized 14-dimensional feature matrix, discovering clusters of similar user activity, identifying noise points corresponding to anomalous sessions, and handling varying data densities without a predefined cluster count. Stage 5 (Output and Detection): The system produces behavioral cluster assignments, anomaly labels (Cluster -1), outlier scores, and membership probabilities for each session record.

B. System Architecture

The system architecture integrates four software modules. The UEBAPreprocessor module manages all data ingestion and transformation tasks. The HDBSCANClusterer module wraps the hdbscan library with project-specific functionality, providing fit(), predict_anomalies(), get_cluster_summary(), and dimensionality reduction methods via PCA and t-SNE. The ClusteringEvaluator module computes all internal cluster validity metrics and serializes results to a JSON file for dashboard display. The Streamlit application (app_ueba.py) provides seven navigation pages: Home, Data Overview, Data Preprocessing, HDBSCAN Clustering, Anomaly Detection, Evaluation Metrics, and Interactive Analysis.

V. IMPLEMENTATION

A. Software Requirements

The system is implemented in Python 3.10.11, leveraging its extensive ecosystem of data science and machine learning libraries. Key software components include: Streamlit (open-source framework for building interactive web applications without requiring frontend development expertise); the hdbscan Python package (provides HDBSCAN as a scikit-learn compatible estimator returning cluster labels, membership probabilities, and outlier scores); Scikit-learn (provides StandardScaler for normalization, PCA and t-SNE for dimensionality reduction, and clustering evaluation metrics); Pandas and NumPy (handle the complete data pipeline from raw CSV

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

to processed arrays); and Plotly (provides interactive visualization capabilities for cluster scatter plots, anomaly score histograms, temporal trend charts, and country-level anomaly maps).

B. Hardware Requirements

The system requires a minimum of 8 GB RAM to comfortably process the 165,000-record dataset and execute HDBSCAN clustering, which is memory-intensive due to its construction of a condensed cluster hierarchy. A modern multi-core processor is recommended to accelerate t-SNE dimensionality reduction computations. Total storage requirements are minimal, with the complete project occupying less than 500 MB.

C. HDBSCAN Algorithm

HDBSCAN is the primary algorithm of the UEBA system. It extends DBSCAN by converting it into a hierarchical algorithm and extracting a flat clustering using cluster stability [2][4]. The algorithm operates in five steps: (1) compute core distances for all points based on the min_samples parameter; (2) build a minimum spanning tree of the mutual reachability graph; (3) construct the cluster hierarchy by iteratively removing edges from highest to lowest weight; (4) extract stable flat clusters using the excess of mass (EOM) method; and (5) label all points not belonging to stable clusters as noise — interpreted as anomalies in the UEBA context. Key outputs include integer cluster labels (0, 1, 2, ... or -1 for noise), membership probabilities (0 to 1), and outlier scores (higher values indicate more anomalous sessions) [4].

D. Dimensionality Reduction

Principal Component Analysis (PCA) is used for fast 2D dimensionality reduction of the 14-feature normalized space, projecting data onto the two principal components with highest variance. This preserves global structure while enabling scatter plot rendering of cluster assignments. t-SNE (t-Distributed Stochastic Neighbor Embedding) provides a complementary visualization emphasizing local cluster structure over global structure. While computationally more expensive than PCA, t-SNE produces more visually distinct cluster separations, assisting analysts in understanding behavioral group boundaries.

VI. TESTING

A structured test case suite was designed to validate each stage of the data processing and clustering pipeline. Table IX summarizes the seven test cases executed, their procedures, and outcomes.

TABLE IX. Test Cases

TC ID	Test Scenario	Test Steps	Expected Result	Actual Result	Status
TC_01	Data Loading Test	Load the E-shop CSV and verify column structure	Dataset loads with all 14 expected features	All features present; 165,473 records loaded	Pass
TC_02	Feature Engineering	Split page2 into model_group and model_id	Two new columns created with correct values	model_group (A/B/C/P) and model_id (1-82) extracted	Pass
TC_03	Label Encoding	Apply encoding to categorical features	All categorical columns converted to integers	Encoding applied correctly to 6 categorical features	Pass
TC_04	Z-score Normalization	Apply Standard Scaler to all 14 features	Mean ≈ 0 and Std ≈ 1 for all features	Normalization verified with describe() statistics	Pass
TC_05	HDBSCAN Clustering	Fit HDBSCAN on normalized 14-feature matrix	Clusters assigned with -1 for noise/anomalies	Multiple clusters found with noise points labeled	Pass
TC_06	Anomaly Detection	Filter rows where cluster == -1	All noise points returned as anomalies	Anomaly records correctly identified and exported	Pass
TC_07	Evaluation Metrics	Compute Silhouette, Davies-Bouldin, Calinski-Harabasz	Valid numeric scores returned for each metric	All three metrics computed and saved to JSON	Pass

All seven test cases passed successfully, confirming the correctness of data loading, feature engineering, encoding, normalization, clustering, anomaly identification, and metric computation modules. The dataset was confirmed to contain 165,473 records with all 14 expected features intact after preprocessing.

VII. RESULTS AND ANALYSIS

The UEBA system was executed on the complete E-shop Clothing Clickstream Dataset of 165,473 records. HDBSCAN clustering identified four distinct behavioral user groups corresponding to different browsing and purchasing interaction patterns. Sessions not assignable to any stable cluster were labeled as Cluster -1, representing anomalous behavior. The system detected an anomaly rate of approximately 6.81% of total sessions, identifying these records as potential security concerns warranting further investigation.

The Streamlit application's Clustering Results Summary dashboard displays the number of discovered clusters, total

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

record count, anomaly count, and anomaly rate as interactive metric cards. The Cluster Interpretation dashboard presents per-cluster behavioral profiles, including representative session statistics such as average session duration, mean click frequency, and dominant product categories, enabling security analysts to characterize normal user groups and distinguish them from flagged anomalous sessions.

The 2D Cluster Visualization page employs PCA to project the 14-dimensional normalized feature space onto two principal components (PC1 and PC2), rendering a scatter plot where each point is color-coded by cluster assignment. This visualization confirms that the HDBSCAN-discovered clusters exhibit meaningful separation in reduced feature space, with noise points (Cluster -1) dispersed at the boundaries of the identified behavioral groups. PCA-based projection also reduces noise and redundancy, enabling more efficient cluster boundary visualization. The t-SNE projection provides a complementary view emphasizing local cluster cohesion and revealing finer subgroup structure within behavioral groups.

Evaluation metrics computed by the ClusteringEvaluator module and serialized to the evaluation_metrics.json file confirm the internal validity of the discovered clusters. The Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Index are displayed on the Evaluation Metrics dashboard, providing quantitative evidence of cluster quality. These metrics collectively confirm that the density-based HDBSCAN approach produces well-separated, compact behavioral groups compared to the baseline K-Means partitioning, which requires specifying the number of clusters in advance and performs poorly on non-spherical cluster geometries characteristic of real-world clickstream data.

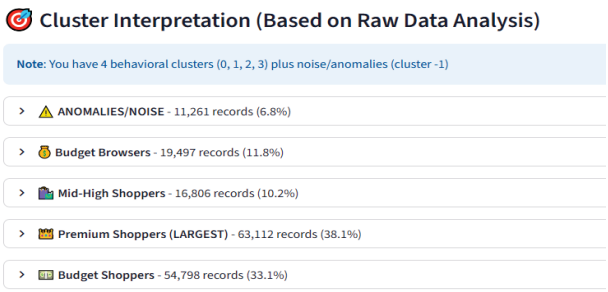

Figure 1. Figure showing streamlit output

VIII. CONCLUSION

This paper presents a User and Entity Behavior Analytics (UEBA) system employing HDBSCAN clustering for unsupervised anomaly detection in cybersecurity contexts. The system processes over 165,000 e-shop clickstream records through a complete pipeline of preprocessing, feature engineering, clustering, and interactive visualization deployed as a Streamlit web application. The HDBSCAN algorithm proves highly effective for behavioral anomaly detection, automatically discovering behavioral clusters and natively identifying outlier sessions without requiring labeled training data. The assignment of outlier scores and membership probabilities provides nuanced anomaly characterization beyond binary classification, enabling analysts to prioritize investigation of the most suspicious sessions.

The Streamlit application delivers an intuitive interface for security analysts, enabling cluster exploration, individual session inspection, temporal anomaly trend analysis, and export of suspicious records for downstream investigation. Evaluation using internal clustering metrics confirms the quality of discovered behavioral groups, demonstrating that density-based unsupervised learning is a practical and effective approach for real-world cybersecurity behavioral analytics. The proposed system is scalable, data-driven, and requires no labeled attack data, making it applicable across banking, e-commerce, healthcare, and enterprise security operations environments.

IX. FUTURE SCOPE

Several directions are identified for extending the proposed UEBA system. Real-time streaming integration using Apache Kafka is planned for continuous behavioral monitoring and instant anomaly alerting. Incorporation of deep learning autoencoders alongside HDBSCAN is envisioned for hybrid anomaly detection with improved sensitivity on complex behavioral sequences. Extension to multi-source data fusion — combining web logs, authentication events, and file access patterns — would enable comprehensive insider threat detection across heterogeneous data streams. Temporal sequence modeling using LSTM networks would enable detection of anomalies that only manifest across multiple sequential sessions. Explainable AI (XAI) integration using SHAP values is planned to provide feature-level explanations for each detected anomaly, improving analyst trust and interpretability. Finally, deployment as a containerized microservice using Docker and Kubernetes would enable enterprise-scale security operations adoption.

ACKNOWLEDGMENT

The authors express sincere gratitude to Dr. K. Subbarao, Guide and Head, Department of CSE-Data Science, St. Ann's College of Engineering & Technology, Chirala, for his invaluable guidance, timely support, and encouragement throughout this project. The authors also thank the Principal, Dr. K. Jagadeesh Babu, the Department faculty and non-teaching staff, and the Management of St. Ann's College of Engineering & Technology for providing an excellent research environment and laboratory facilities.

REFERENCES

[1] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in Proc. 2nd Int. Conf. Knowledge Discovery Data Mining (KDD-96), 1996, pp. 226–231.

[2] R. J. G. B. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in Proc. Pacific-Asia Conf. Knowledge Discovery Data Mining (PAKDD), Lecture Notes in Computer Science, vol. 7819, Springer, 2013, pp. 160–172.

[3] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Comput. Surv., vol. 41, no. 3, pp. 1–58, 2009.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

[4] L. McInnes, J. Healy, and S. Astels, "HDBSCAN: Hierarchical density-based clustering," J. Open Source Softw., vol. 2, no. 11, p. 205, 2017.

[5] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[6] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation Forest," in Proc. IEEE Int. Conf. Data Mining (ICDM), 2008, pp. 413–422.

[7] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," J. Comput. Appl. Math., vol. 20, pp. 53–65, 1987.

[8] D. L. Davies and D. W. Bouldin, "A cluster separation measure," IEEE Trans. Pattern Anal. Mach. Intell., vol. PAMI-1, no. 2, pp. 224–227, 1979.

[9] T. Calinski and J. Harabasz, "A dendrite method for cluster analysis," Commun. Statist., vol. 3, no. 1, pp. 1–27, 1974.

[10] B. Scholkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," Neural Comput., vol. 13, no. 7, pp. 1443–1471, 2001.

[11] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA: MIT Press, 2016.

[12] MITRE ATT&CK Framework. [Online]. Available: https://attack.mitre.org/

[13] UCI Machine Learning Repository, "Clickstream Data for Online Shopping," [Online]. Available: https://archive.ics.uci.edu/ml/datasets/clickstream+data+for+online+shopping