Global Scientific JOURNALS

GSJ: Volume 13, Issue 4, April 2025, Online: ISSN 2320-9186
www.globalscientificjournal.com

USER BEHAVIOUR ANOMALY DETECTOR

HARI KRISHNAN.B
Department of Computer science and engineering
Dr.M.G.R. Educationaland Research Institute Deemed to be University.
Harikrishnan.cfis@gmail.com

SURESHKUMAR.R
Department of Computer science and engineering
Dr.M.G.R. Educationaland Research Institute Deemed to be University.
sureshkumar.cfis@gmail.com

VISHWA.S
Department of Computer science and engineering
Dr.M.G.R. Educationaland Research Institute Deemed to be University.
s.vishwa.cfis@gmail.com

Dr.S Mohandoss
Associate professor
Department of Computer science and engineering
Dr.M.G.R.Educationaland Research Institute
Deemed to be University

Ms.MagnaYadlapalli
Assistant professor
Faculty of Center of Excellence in Digital Forensic

Abstract

The User Behavior Anomaly Detector is an extensive software developed to identify abnormal patterns of user behavior with machine learning and real-time processing. The system uses Isolation Forest, an unsupervised machine learning algorithm, for the detection of anomalies in user behavior data. Flask is used as the backend for web interaction; SQLite is employed in the data storage for user behavior; PyWebview GUI enables a comfortable desktop-like interaction, whereas FastAPI is used for scalable API-based interaction and testing.The guardians of the system shall enter behavior data through a web interface, and the Isolation Forest model will come into play for computation.

On the detection of an anomaly, the system is also supposed to send an email alert to notify administrators. There is a dynamic Chart.js display to observed user behavior data over time in the application, so the users can visualize these events over time among general trends. Thereafter, CRUD operations are performed in the backend to allow data retrieval and storage for maintaining logs of very healthy dynamics for user behavior. To make it more scalable and

averse to abuse, FastAPI and rate limiting middleware are used in the project to prevent abuse. To encapsulate makes modular system that would be easier to scale further in the future; for example, incorporate more machine learning models or add to the API for external use. User Behavior Anomaly Detector, in total, is a strong real-time anomaly detection, easy-to-use tool usable for cybersecurity, fraud detection, and user activity monitoring

Keywords: Anomaly Detection, Machine Learning, Isolation Forest, Cybersecurity, Real-Time Processing.

I. INTRODUCTION

The User Behavior Anomaly Detector is a comprehensive software solution engineered to identify abnormal patterns in user behavior by leveraging advanced machine learning techniques and real-time data processing. At its core, the system utilizes the Isolation Forest algorithm—an unsupervised machine learning method renowned for its efficiency in isolating outliers without the need for pre-labeled data—to analyze user activity logs and pinpoint deviations from established norms. This capability is crucial for detecting potential cybersecurity threats, fraudulent activities, or other anomalies that may compromise system integrity.

The architecture of the project is thoughtfully designed to ensure both functionality and scalability. The backend is powered by Flask, which handles web-based interactions, making it straightforward for system guardians to input behavior data through an intuitive web interface. This data is securely stored in an SQLite database, chosen for its lightweight and reliable performance in managing structured logs. To enhance user experience, a PyWebview GUI is integrated, providing a desktop-like environment that simplifies navigation and monitoring for administrators.

To further support the system's scalability and modern integration needs, FastAPI is employed for creating a robust API layer. This allows for seamless interaction between different components of the system as well as potential external applications. The API is fortified with rate limiting middleware, ensuring that the system remains resilient against potential abuses such as denial-of-service attacks or other forms of excessive request flooding. This approach not only safeguards the system but also ensures that it can scale efficiently to handle increased loads as the volume of user behavior data grows.

A distinctive feature of the User Behavior Anomaly Detector is its dynamic visualization capability, achieved through the integration of Chart.js. This tool enables real-time tracking and visualization of user behavior trends, allowing administrators to observe changes over time and gain insights into both typical and atypical user activities. Furthermore, once an anomaly is detected, the system is programmed to trigger an email alert, ensuring that relevant stakeholders are immediately notified of potential issues. Additionally, CRUD operations are implemented in the backend to facilitate effective data retrieval and the maintenance of detailed logs that document the system's performance and the overall health of user interactions.

In summary, the User Behavior Anomaly Detector represents a state-of-the-art solution in the realm of cybersecurity and fraud detection. By combining the analytical power of the Isolation Forest algorithm with a robust, modular architecture that includes Flask, FastAPI, SQLite, PyWebview, and Chart.js, the system provides a reliable and scalable platform for real-time anomaly detection. This design not only ensures immediate responsiveness to potential threats but also lays the groundwork for future enhancements—such as the integration of additional machine learning models or expanded API functionalities—making it an invaluable tool for monitoring and securing digital environments.

METHODOLOGY

The User Behavior Anomaly Detector employs a sophisticated methodology to identify unsupervised anomalous activities by analyzing user behavior through their IP addresses. This process integrates advanced machine learning techniques with real-time data processing to ensure accurate and timely detection of irregular patterns.

Data Collection and Preprocessing

The initial phase involves the systematic collection of user behavior data, focusing on IP addresses, timestamps, accessed resources, and interaction patterns. This data is meticulously preprocessed to handle missing values, eliminate redundancies, and standardize formats, thereby ensuring its suitability for subsequent analysis.

Feature Extraction

Post-preprocessing, the system extracts pertinent features that encapsulate user behavior. These features may include session durations, frequency of resource access, data transfer volumes, and geolocation information derived from IP addresses. Such comprehensive feature extraction facilitates a nuanced understanding of typical user behavior patterns.

Isolation Forest Algorithm Implementation

The core analytical engine of the system is the Isolation Forest algorithm, an unsupervised machine learning method adept at anomaly detection. Unlike traditional models that rely on profiling normal behavior, the Isolation Forest operates by isolating data points. It constructs random decision trees (iTrees) by recursively partitioning the data based on randomly selected features and split values. The fundamental premise is that anomalies, being few and distinct, are more susceptible to isolation in fewer partitioning steps compared to normal instances.

Anomaly Scoring Mechanism

Each data point's path length within the iTrees is recorded, representing the number of partitions required to isolate the point. Anomalous points typically exhibit shorter path lengths due to their distinctiveness. The system calculates an anomaly score for each point, with scores approaching 1 indicating a higher likelihood of anomaly. This scoring mechanism enables the differentiation between normal and aberrant user behaviors.

Real-Time Detection and Alerting

The system continuously monitors incoming user behavior data, applying the Isolation Forest model to assess anomaly scores in real-time. Upon detecting an anomaly—such as an IP address exhibiting unusual access patterns or attempting unauthorized resource access—the system promptly triggers an alert mechanism. This typically involves sending immediate notifications to administrators, enabling swift investigation and mitigation of potential security threats.

Visualization and Reporting

To facilitate comprehensive monitoring, the system incorporates dynamic visualization tools, such as Chart.js, to depict user behavior trends over time. Administrators can observe patterns, identify deviations, and analyze the context of anomalies through interactive dashboards. Additionally, the system maintains detailed logs of detected anomalies, supporting in-depth forensic analysis and reporting.

Scalability and Abuse Prevention

The architecture is designed for scalability, employing FastAPI for efficient API-based interactions and integrating rate limiting middleware to prevent abuse. This ensures the system can handle increasing data volumes and user requests without compromising performance or security.

Modular Design for Future Enhancements

The modular design of the system facilitates future enhancements, such as integrating additional machine learning models or extending API functionalities for external applications. This adaptability ensures the system remains robust and relevant in the face of evolving cybersecurity challenges.

In summary, by meticulously analyzing user behavior through IP addresses using the Isolation Forest algorithm, the User Behavior Anomaly Detector provides a robust framework for real-time detection of unsupervised anomalous activities, thereby enhancing the security and integrity of digital environments.

EXISTING SYSTEM

Existing systems for detecting anomalies in user behavior encompass a range of sophisticated methodologies and technologies, each tailored to identify and mitigate potential threats within digital environments. One notable example is PRODIGAL (Proactive Discovery of Insider Threats Using Graph Analysis and Learning), a system developed under DARPA's Anomaly Detection at Multiple Scales (ADAMS) project. Established in 2011 with a budget of $9 million, PRODIGAL leverages graph theory, machine learning, statistical anomaly detection, and high-performance computing to analyze vast datasets, including network traffic, emails, text messages, and server logs. By processing terabytes of data daily, it aims to proactively identify insider threats by detecting anomalous patterns indicative of malicious activities.

Another prominent system is Guardian Analytics, acquired by NICE Actimize in 2020. This company specializes in behavioral analytics and machine learning technologies to prevent banking fraud. Their products utilize anomaly detection techniques to monitor financial transactions, creating probabilistic profiles of individual users to identify deviations from typical behavior. As of September 2016, nearly 430 financial institutions relied on Guardian Analytics' services to mitigate fraud risk and counter sophisticated criminal attacks.

In the realm of privacy-preserving anomaly detection, the concept of Local Differential Privacy (LDP) has gained traction. LDP ensures that individual data points are obfuscated before analysis, protecting user privacy while still allowing for effective anomaly detection. This approach is particularly relevant in social networks, where preserving user privacy is paramount. For instance, a model utilizing restricted LDP sanitizes collected user information to create synthetic data copies. These reconstructed datasets enable the classification of user activity and the detection of abnormal network behavior without compromising individual privacy.

Collectively, these systems exemplify the diverse strategies employed in existing user behavior anomaly detection frameworks, ranging from large-scale data mining and graph analysis to privacy-preserving methodologies, all aimed at enhancing security and mitigating potential threats.

SYSTEM IMPLEMENTATION

The User Behavior Anomaly Detector is meticulously designed to ensure seamless and efficient operation through the integration of several key components. At the forefront is the User Interface, which serves as the primary point of interaction for users, enabling them to input data and receive feedback in a user-friendly environment. This interface communicates directly with the Backend Server, responsible for handling user requests, processing data, and orchestrating the system's core functionalities. Central to the system's analytical capabilities is the Isolation Forest Model, an unsupervised machine learning algorithm adept at detecting anomalies within user behavior data by identifying patterns that deviate from the norm. All user behavior data is securely stored in the Database, ensuring that historical information is readily available for analysis and future reference. Upon the detection of any anomalies, the Notification System is activated to promptly send email alerts to administrators, facilitating immediate attention to potential issues. To provide insights into user activities over time, the system incorporates a Visualization component, which presents historical data through dynamic charts, allowing for intuitive monitoring and analysis. Finally, the system's Testing & Scalability measures are rigorously implemented to ensure robustness and the capacity to efficiently handle growth, maintaining optimal performance as user demands evolve.

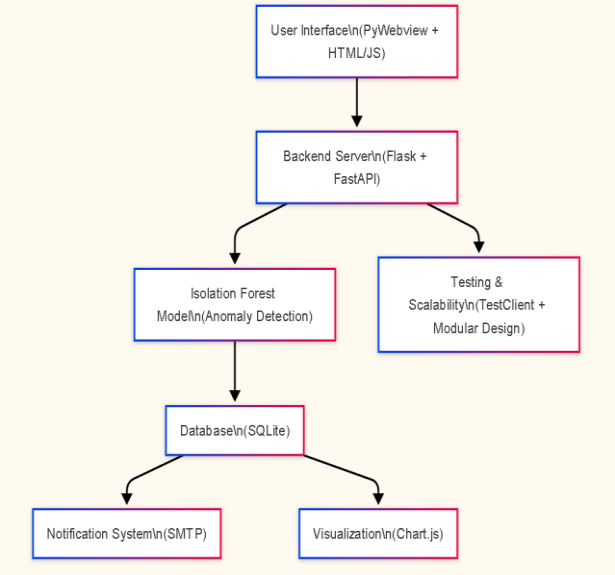

DIAGRAM EXPLANATION

User Interface (UI)

The UI serves as the main interaction point for users, providing an intuitive dashboard. It allows users to view data, receive alerts, and input necessary parameters for analysis. A well-designed UI enhances usability, ensuring seamless system navigation.

Backend Server

The backend processes requests, manages data flow, and ensures system functionality. It interacts with the database and the anomaly detection model to deliver accurate results. Security, efficiency, and scalability are key factors in backend development.

Isolation Forest Model

This machine learning model detects anomalies by identifying outliers in user behavior. It isolates abnormal data points more efficiently than normal data, ensuring accuracy. Used in fraud detection, cybersecurity, and network monitoring applications.

Database

Stores and manages user behavior data for future analysis and reference. Ensures data integrity, security, and fast retrieval for anomaly detection. Common databases used include PostgreSQL, MySQL, and MongoDB.

Notification System

Sends real-time alerts via email or SMS when anomalies are detected.Helps administrators and users take immediate action on suspicious activities .Integrated using services like SendGrid, AWS SES, or SMTP protocols.

Visualization

Displays user behavior data using dynamic charts and graphs for analysis. Helps users track patterns, trends, and anomalies over time.Common visualization tools include Chart.js, D3.js, and Plotly.

Testing & Scalability

Ensures the system is robust, secure, and performs well under high traffic Includes unit testing, load testing, and security assessments for reliability. Cloud deployment and microservices enhance scalability and performance.

CONCLUSION

In conclusion, the User Behavior Anomaly Detector exemplifies a comprehensive and robust solution for monitoring and analyzing user activities to identify abnormal patterns indicative of potential security threats. By integrating the Isolation Forest algorithm, an unsupervised machine learning technique, the system effectively detects anomalies without the need for labeled datasets, enhancing its adaptability to various environments. The utilization of Flask for backend web interactions, SQLite for data storage, PyWebview for a seamless desktop-like GUI, and FastAPI for scalable API interactions collectively contribute to a modular and efficient architecture. The system's capability to send real-time email alerts upon anomaly detection ensures prompt administrative responses, while dynamic data visualization through Chart.js facilitates intuitive monitoring of user behavior trends. Furthermore, the implementation of CRUD operations and rate-limiting middleware underscores the system's commitment to data integrity, scalability, and abuse prevention. Overall, the User Behavior Anomaly Detector stands as a valuable tool in cybersecurity, fraud detection, and user activity monitoring, offering a scalable and user-friendly approach to safeguarding digital environments.

REFERENCES

1. Gupta and r. Sharma, "securewifi: A Browser Extension for Encrypted Data Transmission over Public WiFi Networks," IEEE Transactions on Information Forensics and Security, vol. 19, no. 1, pp. 234-248, 2024.
2. M. Rahim, S. Khan, and J. Lee"PrivacyGuard:AComprehensive Browser Extension for Data Privacy on Public WiFi," IEEE Access, vol. 11, pp. 67890- 67905, 2023.
3. P. Bhattacharya and K. Srinivasan, "WiFiShield: Enhancing Browser Security for Public WiFi

Networks," IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 3, pp. 1456-1470, 2022

4. T. Wang, Y. Li, and X. Chen, "SecureBrowse: A Browser Extension for Secure Communication over Untrusted Networks," IEEE Transactions on Mobile Computing, vol. 21, no. 6, pp. 2056-2070, 2021.
5. J. Kim, H. Park, and Y. Choi, "PrivacyShield: A Browser Extension for Protecting User Privacy on Public WiFi Networks," IEEE Internet of Things Journal, vol. 8, no. 4, pp. 2876-2889, 2020.