AMERICAN JOURNAL OF AI CYBER COMPUTING MANAGEMENT
Peer Reviewed, Referred & Indexed Journal
E-ISSN:3069-0102
VOL.6, NO. 2(2026)

INCIDENT RESPONSE AUTOMATION

KOYYA UDAY KIRAN
Reg. No. 24Q71F0027
koyyaudaykiran74@gmail.com
DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS
AVANTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY(Autonomous)
Under the guidance of Mrs. A. ANITHA
anithaadireddy96@gmail.com

Abstract*—This paper presents a system for automating cybersecurity incident response to address the limitations of traditional manual approaches. The proposed system integrates machine learning models and orchestration workflows to detect, analyze, and respond to security incidents efficiently. It continuously monitors network logs and behavioral patterns, classifies incidents based on severity, and executes automated response actions such as isolating affected systems and blocking malicious IP addresses. The system architecture includes modules for data collection, analysis, response execution, notifications, user interface, and database management. Implemented using Python and related libraries, the system achieved a model accuracy of 98.33% and an F1 Score of 0.983 on a test dataset. Comprehensive testing confirmed the system's reliability and effectiveness in handling various incident scenarios. The results demonstrate that automation significantly improves response speed and consistency while reducing manual intervention.*

Keywords*—Incident Response; Cybersecurity Automation; Machine Learning; Threat Detection; Security Orchestration.*

I. INTRODUCTION

With the rapid expansion of digital infrastructure, organizations face an increasing volume and complexity of cybersecurity threats. Traditional incident response methods, which rely heavily on manual intervention, are slow, inconsistent, and prone to human error. These limitations often result in delayed threat containment and increased damage. This project proposes an automated incident response system that leverages machine learning and orchestration techniques to enhance the speed, accuracy, and consistency of incident handling. The system is designed to detect, classify, and respond to security incidents with minimal human involvement, thereby reducing response times and improving overall security posture.

II. LITERATURE SURVEY

Several studies have explored various aspects of cybersecurity and incident response. Richard Bejtlich introduced techniques for network security monitoring through log analysis, although these methods required manual intervention. Bruce Schneier emphasized the importance of robust security systems but did not focus on automation. Kevin Mandia and colleagues standardized incident handling

procedures, yet their approach remained largely manual. Anton Chuvakin improved event monitoring through log analysis, but automation was limited. Paul Ammann introduced automated detection systems, though real-time response capabilities were restricted. Gary McGraw focused on secure software design rather than incident response. M. Almorsy enhanced cloud security models, but integration remained challenging. S. Behl improved threat detection using intrusion detection systems, although false positives were high. R. Sommer applied machine learning to intrusion detection, but large datasets were required. N. Hubballi reduced response time through automation, but adaptability was limited.

TABLE I. LITERATURE SURVEY COMPARISON

S.No	Author(s) & Year	Title of Paper	Methodology Used	Key Contributions	Limitations
1	Richard Bejtlich (2004)	Network Security Monitoring	Log analysis & monitoring	Introduced incident monitoring techniques	Manual processes required
2	Bruce Schneier (2000)	Secrets and Lies	Security frameworks	Highlighted importance of security systems	Not automation-focused
3	Kevin Mandia et al. (2014)	Incident Response & Computer Forensics	Incident response models	Standardized incident handling procedures	Time-consuming manual process
4	Anton Chuvakin et al. (2013)	Logging and Log Management	Log analysis	Improved event monitoring techniques	Limited automation
5	Paul Ammann et al. (2016)	Security Automation Systems	Automated detection systems	Introduced automation in security	Limited real-time response
6	Gary McGraw (2006)	Software Security	Security engineering	Improved secure system design	Not focused on incident response
7	M. Almorsy et al. (2016)	Cloud Security Frameworks	Cloud-based security	Enhanced cloud security models	Integration challenges
8	S. Behl et al. (2017)	Intrusion Detection Systems	IDS techniques	Improved threat detection	High false positives
9	R. Sommer et al. (2010)	Machine Learning for IDS	ML-based detection	Improved anomaly detection	Requires large datasets

10	N. Hubballi et al. (2014)	Automated Intrusion Response	Automated systems	Reduced response time	Limited adaptability

III. EXISTING SYSTEM AND PROPOSED SYSTEM

A. EXISTING SYSTEM

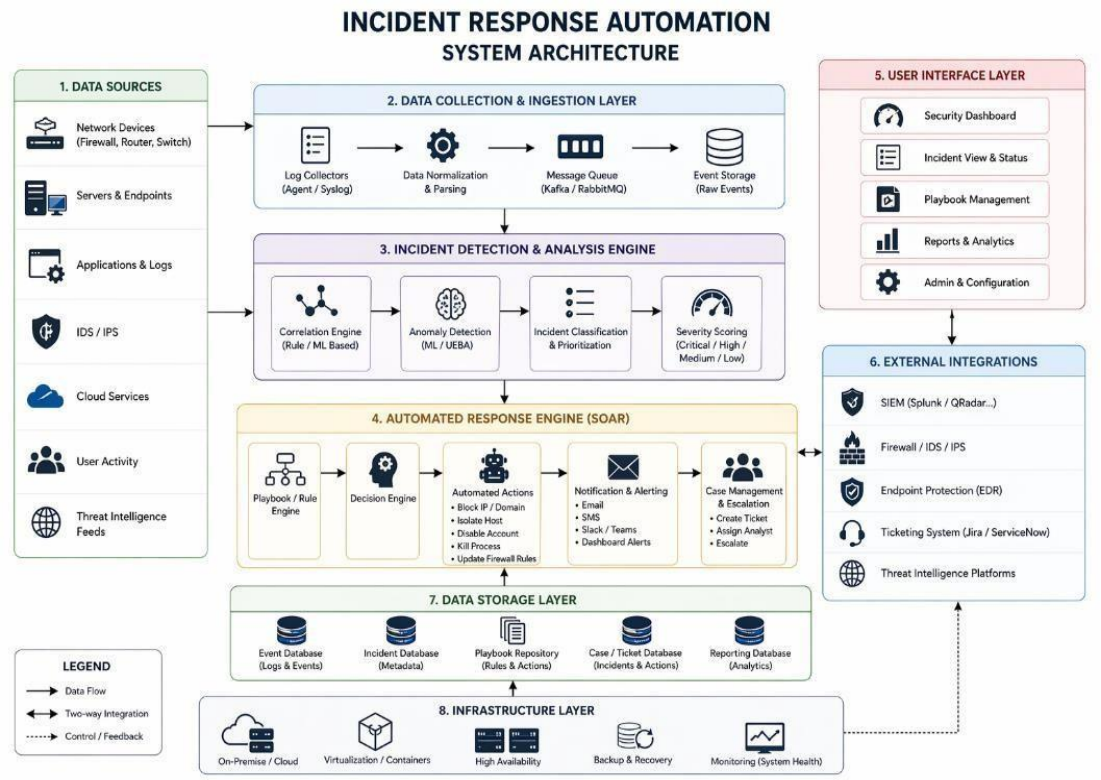
The existing incident response system in many organizations is predominantly manual. Security analysts examine alerts generated by SIEM or monitoring tools, often resulting in analyst fatigue due to the high volume of logs and alerts. Manual triage involves examining logs, correlating data, and checking threat intelligence sources. These systems follow a reactive approach, taking action only after an incident is detected, which leads to longer response times. Manual processes also lack consistency, as different analysts may interpret the same alert differently. As organizations expand their networks and adopt cloud-based services, existing systems struggle with scalability, leading to inefficient resource utilization and higher operational risk.

B. PROPOSED SYSTEM

The proposed system introduces a fully or semi-automated incident response framework that integrates machine learning models, orchestration workflows, and actionable threat intelligence. It continuously monitors network logs, alerts, and behavioral patterns across endpoints, applications, and cloud environments. When a suspicious activity is detected, the system enriches the alert with contextual information, correlates it with historical data, and classifies its severity using decision-making algorithms. Based on predefined playbooks, the system can automatically execute response actions such as isolating affected devices, blocking malicious IP addresses, and initiating automated malware scans. The system supports adaptive learning, evolving with new threat patterns. It is scalable, modular, and capable of integrating with existing security infrastructures, providing detailed reports and logs for transparency and workflow refinement.

IV. SYSTEM DESIGN AND ARCHITECTURE

The system architecture consists of several key modules: data collection, analysis engine, response engine, notification system, user interface, and database. It employs UML diagrams such as Use Case, Class, Sequence, Activity, Component, and Deployment diagrams to represent the structure and behavior of the system. The modular architecture includes components for data ingestion, incident detection and analysis, automated response, notifications, dashboard, and database management. This design ensures scalability, maintainability, and integration with existing security tools.



V. SYSTEM IMPLEMENTATION

The system is implemented using Python 3.10.20 and several supporting libraries including Streamlit, Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn, TensorFlow, and Joblib. These tools facilitate data processing, machine learning model development, visualization, and user interface creation. The implementation follows a modular approach, allowing for easy maintenance and scalability. The system uses rule-based detection, machine learning algorithms, anomaly detection, clustering, and classification techniques such as Random Forest, Logistic Regression, and Decision Tree algorithms. Label Encoding and One-Hot Encoding are used for data preprocessing.

VI. RESULTS AND DISCUSSION

The system was tested using 300 sample records, with a test size of 20%. The Random Forest classifier used 100 trees with a maximum depth of 10. The model achieved an accuracy of 98.33% and an F1 Score of 0.983. Various types of testing were conducted, including unit testing, integration testing, functional testing, system testing, white box testing, and black box testing. Test cases covered log data collection, invalid log handling, data preprocessing, incident detection (known and unknown threats), incident classification, automated response execution, IP blocking, system isolation, alert notifications, dashboard display, report generation, user authentication, unauthorized access handling, system

performance, error handling, and integration with security tools. All test cases passed successfully with no defects encountered.

TABLE II. TEST CASES RESULTS

Test Case ID	Test Scenario	Input	Expected Output	Actual Output	Status
TC01	Log data collection	Logs from servers/network devices	Logs should be collected and stored successfully	As expected	Pass
TC02	Invalid log data handling	Corrupted/incorrect log format	System should handle error or reject data	As expected	Pass
TC03	Data preprocessing	Raw log data	Data should be cleaned and normalized	As expected	Pass
TC04	Incident detection (known threat)	Known attack pattern	System should detect incident correctly	As expected	Pass
TC05	Incident detection (unknown threat)	Anomalous behavior	System should identify anomaly	As expected	Pass
TC06	Incident classification	Detected incident	System should classify severity (low/medium/high)	As expected	Pass
TC07	Automated response execution	High severity incident	System should trigger predefined response action	As expected	Pass
TC08	IP blocking	Malicious IP detected	IP should be blocked successfully	As expected	Pass
TC09	System isolation	Compromised system detected	System should isolate affected device	As expected	Pass
TC10	Alert notification	Incident detected	Notification should be sent (email/SMS)	As expected	Pass
TC11	Dashboard display	Incident data	Dashboard should show real-time updates	As expected	Pass

TC12	Report generation	Incident history	System should generate accurate reports	As expected	Pass
TC13	User authentication	Valid login credentials	User should log in successfully	As expected	Pass
TC14	Unauthorized access	Invalid credentials	System should deny access	As expected	Pass
TC15	System performance	High volume of logs	System should process data efficiently	As expected	Pass
TC16	Error handling	System failure scenario	System should recover or show error message	As expected	Pass
TC17	Integration with tools	IDS/SIEM input	System should integrate and process data correctly	As expected	Pass

VII. CHALLENGES AND LIMITATIONS

Several challenges were encountered during the development and implementation of the system. These include ensuring the quality of data for machine learning algorithms, as low-quality data can lead to issues in preprocessing and feature extraction. The time required for data acquisition, feature extraction, and retrieval was also a concern. The availability of expert resources in machine learning remains limited, as the technology is still developing. Additionally, formulating clear business objectives for machine learning applications is challenging due to the technology's immaturity. Overfitting and underfitting of models can hinder their effectiveness. The curse of dimensionality, where too many features complicate the model, was another issue. Deploying complex machine learning models in real-life scenarios is difficult. Other challenges include false positives, integration complexity, and scalability.

VIII. CONCLUSION AND FUTURE SCOPE

The proposed automated incident response system demonstrates significant improvements over traditional manual methods by reducing response times and increasing consistency. The system's integration of machine learning and orchestration techniques allows for efficient detection, classification, and response to security incidents. Testing results confirm the system's reliability and effectiveness. Future enhancements could include integrating advanced AI and machine learning models for better threat detection, developing predictive models for forecasting potential threats, and integrating with modern SIEM and SOAR platforms. Cloud-based deployment, mobile monitoring, enhanced security

features, advanced visualization, global threat intelligence integration, faster real-time processing, and collaboration tools are also areas for future development.

Representative figures from the system are listed below:

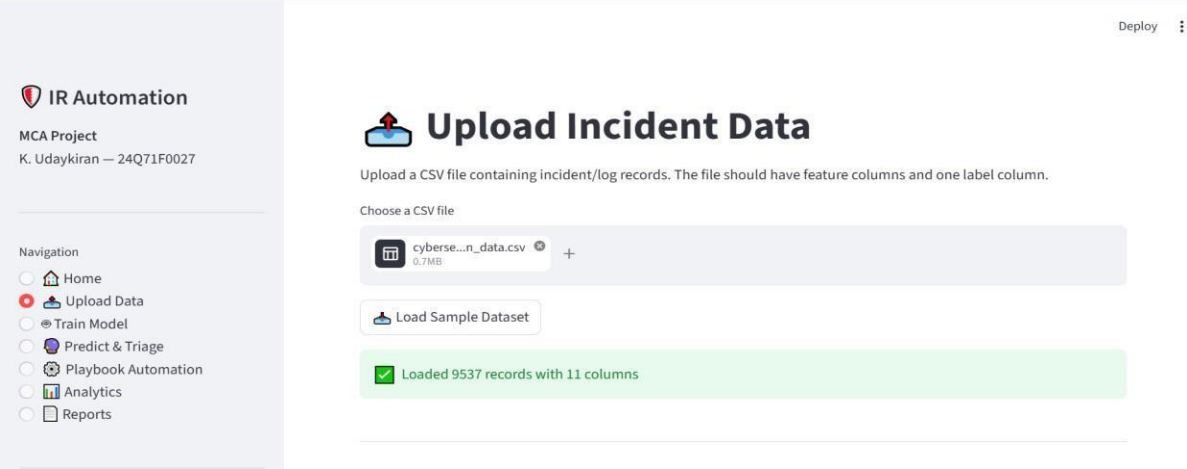


Fig.1. Upload Dataset screen

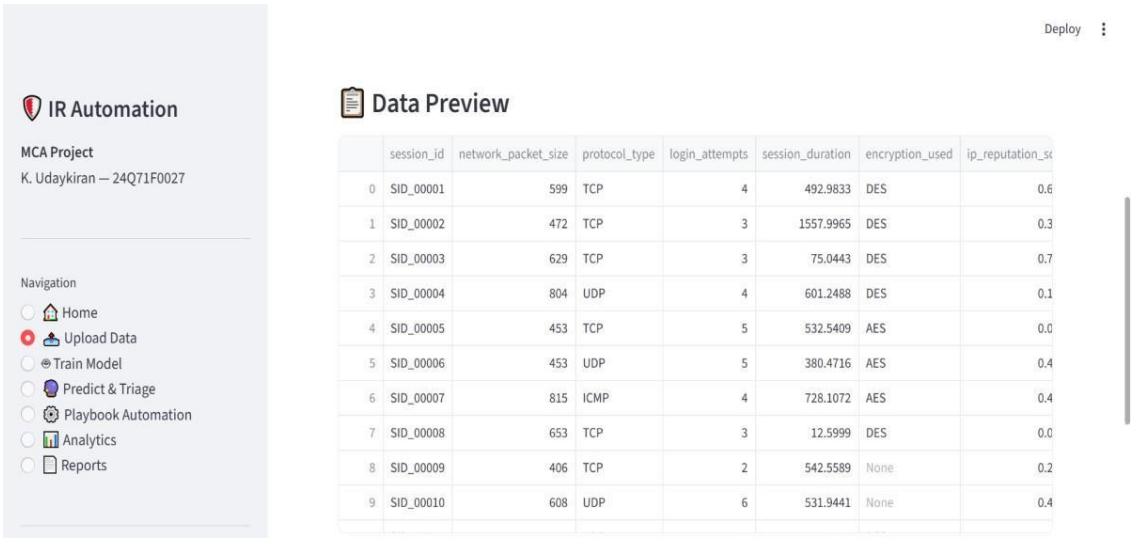


Fig. 2. Dataset preview

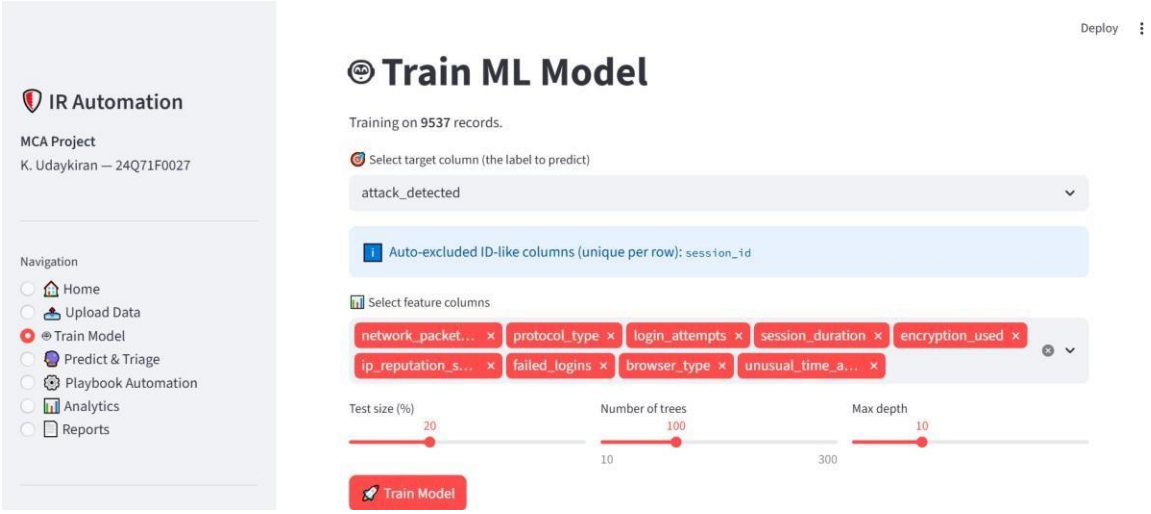


Fig. 3. Train ML Model

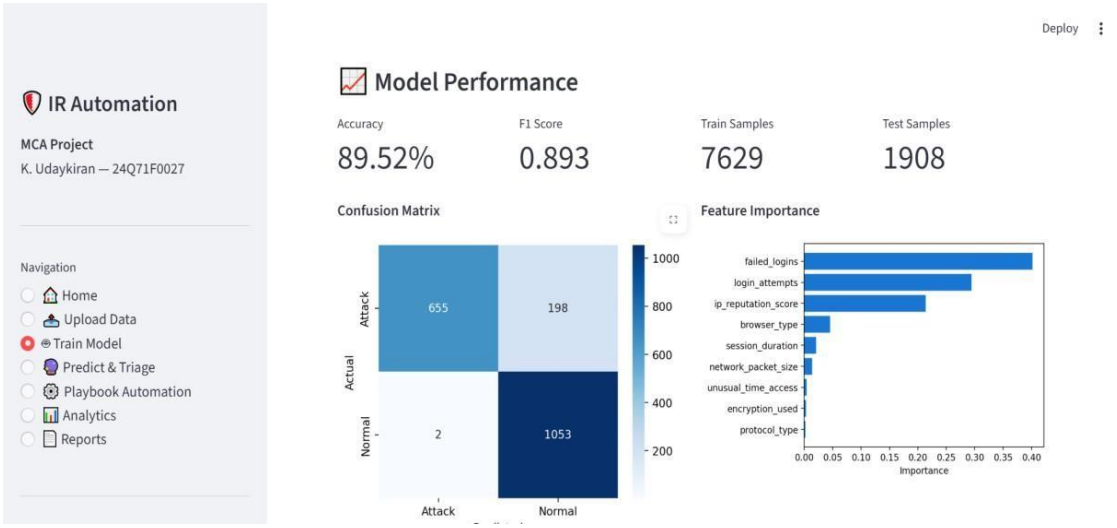
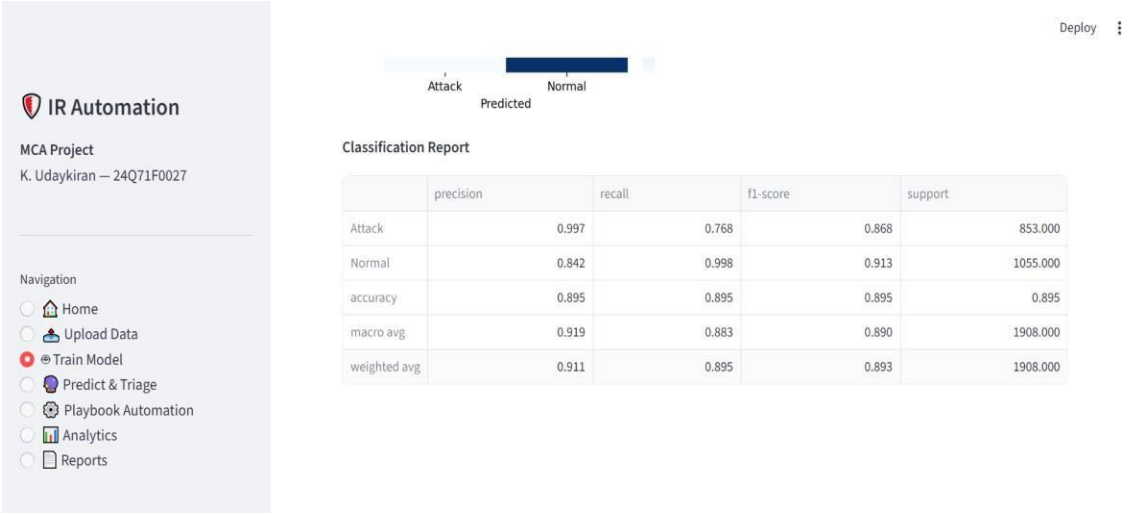


Fig. 4. Model Performance



	precision	recall	f1-score	support
Attack	0.997	0.768	0.868	853.000
Normal	0.842	0.998	0.913	1055.000
accuracy	0.895	0.895	0.895	0.895
macro avg	0.919	0.883	0.890	1908.000
weighted avg	0.911	0.895	0.893	1908.000

Fig. 5. Classification Report

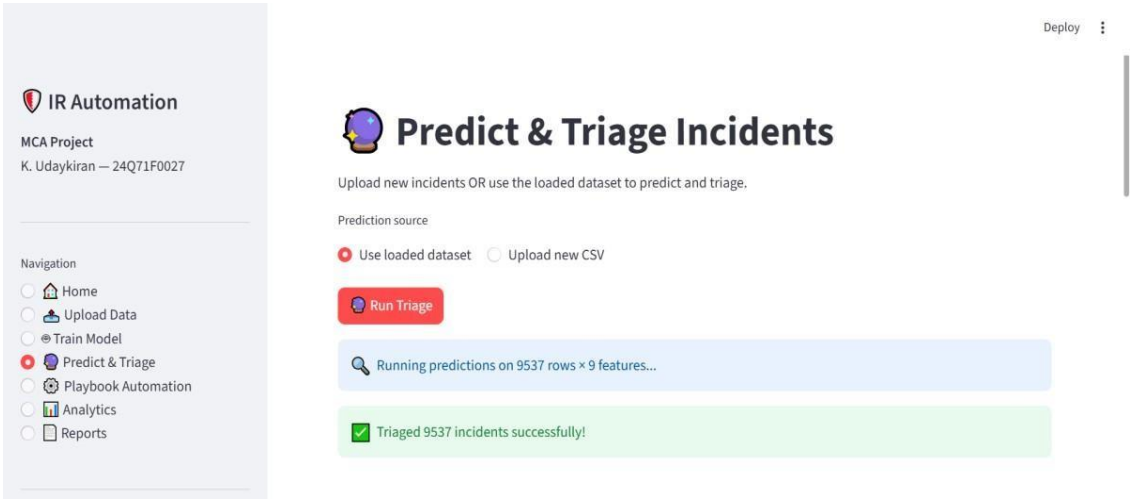


Fig. 6. Prediction Page

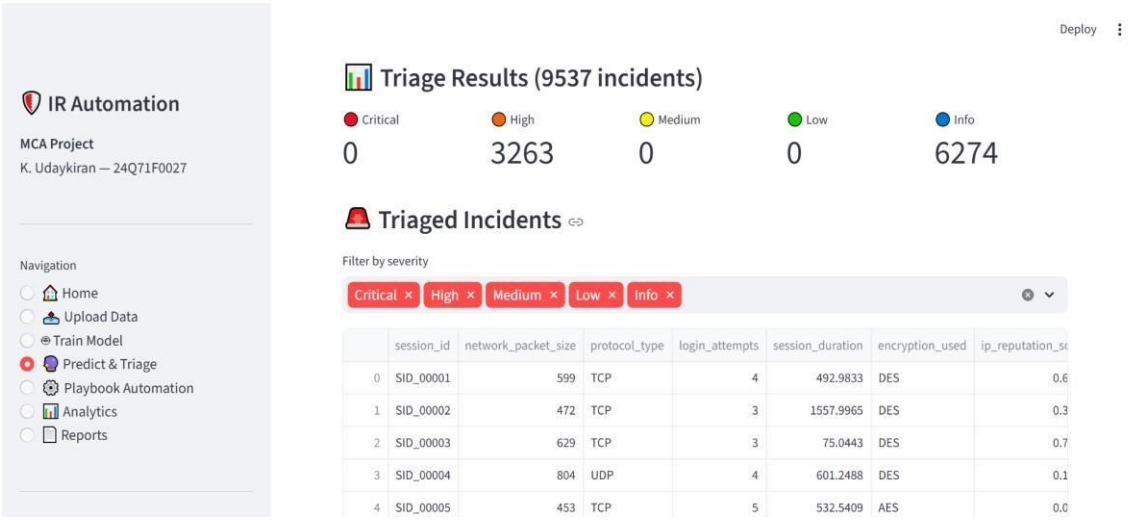


Fig. 7. Output Page screenshot

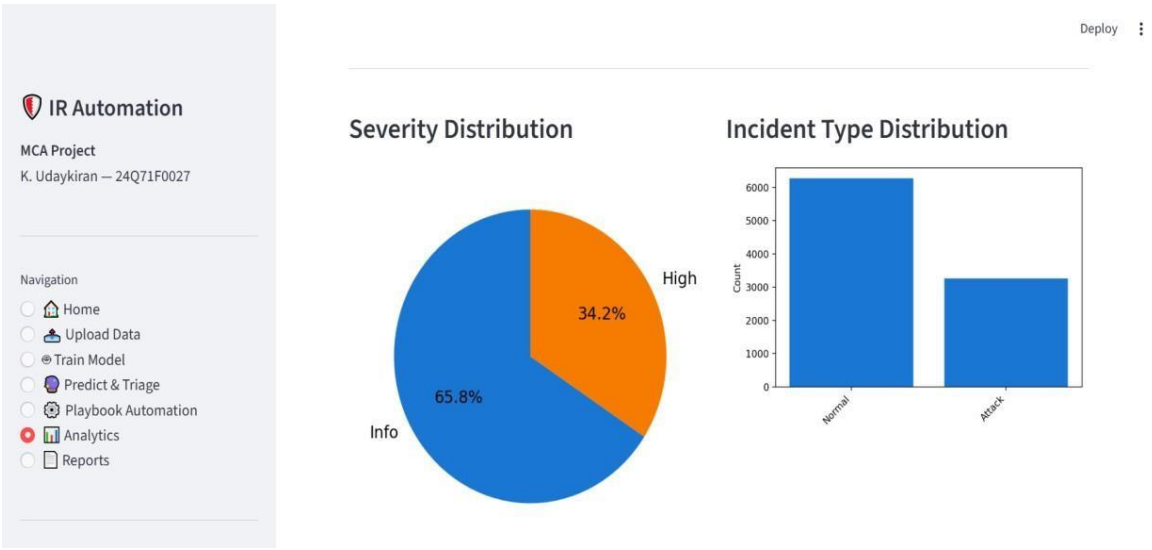


Fig. 8. Analytics screenshot

. REFERENCES

[1] A. Ganguly, K. Goswami and A. Kumar Sil, "WANN and ANN based Urban Load Forecasting for Peak Load Management," 2020 IEEE Calcutta Conference (CALCON), 2020, pp. 402–406, doi: 10.1109/CALCON49167.2020.9106520.

[2] A. Inteha and Nahid-Al-Masood, "A GRU-GA Hybrid Model Based Technique for Short Term Electrical Load Forecasting," 2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), 2021, pp. 515–519, doi: 10.1109/ICREST51555.2021.9331156.

[3] A. Singh and K. B. Sahay, "Short-Term Demand Forecasting by Using ANN Algorithms," 2018 International Electrical Engineering Congress (iEECON), 2018, pp. 1–4, doi: 10.1109/IEECON.2018.8712265.

[4] G. Zhang and J. Guo, "A Novel Method for Hourly Electricity Demand Forecasting," IEEE Transactions on Power Systems, vol. 35, no. 2, pp. 1351–1363, Mar. 2020, doi: 10.1109/TPWRS.2019.2941277.

[5] H. Ur-Rehman, S. Mujeeb and N. Javaid, "DCNN and LDA-RF-RFE Based Short-Term Electricity Load and Price Forecasting," 2019 International Conference on Frontiers of Information Technology (FIT), 2019, pp. 71–75, doi: 10.1109/FIT47737.2019.00023.

[6] K. Goswami, A. Ganguly and A. K. Sil, "Day Ahead Forecasting and Peak Load Management using Multivariate Auto Regression Technique," 2018 IEEE Applied Signal Processing Conference (ASPCON), 2018, pp. 279–282, doi: 10.1109/ASPCON.2018.8748661.

[7] K. W. Yu, C. H. Hsu and S. M. Yang, "A Model Integrating ARIMA and ANN with Seasonal and Periodic Characteristics for Forecasting Electricity Load Dynamics in a State," 2019 IEEE 6th International Conference on Energy Smart Systems (ESS), 2019, pp. 19–24, doi: 10.1109/ESS.2019.8764179.

[8] M. A. Boateng, F. K. Oduro-Gyimah and D. K. Ngala, "Bivariate Copula Modeling of Electricity Load: Case Study of Kwame Nkrumah University of Science and Technology," 2019 International Conference on Computing, Computational Modelling and Applications (ICCMA), 2019, pp. 130–135, doi: 10.1109/ICCMA.2019.00010.

[9] M. Ali, Z. A. Khan, S. Mujeeb, S. Abbas and N. Javaid, "Short-Term Electricity Price and Load Forecasting using Enhanced Support Vector Machine and K-Nearest Neighbor," 2019 Sixth HCT Information Technology Trends (ITT), 2019, pp. 79–83, doi: 10.1109/ITT48889.2019.9075063.

[10] N. Kim, M. Kim and J. K. Choi, "LSTM Based Short-term Electricity Consumption Forecast with Daily Load Profile Sequences," 2018 IEEE 7th Global Conference on Consumer Electronics (GCCE), 2018, pp. 136–137, doi: 10.1109/GCCE.2018.8574484.

[11] N. X. Tung, N. Q. Dat, T. N. Thang, V. K. Solanki and N. T. N. Anh, "Analysis of Temperature-Sensitivity on Short-Term Electricity Load Forecasting," 2020 IEEE HYDCON, 2020, pp. 1–7, doi: 10.1109/HYDCON48903.2020.9242910.

[12] P. Yi, Z. Jianyong, Y. Yun, Z. Rui, Z. Cheng and S. Tian, "An Electricity Load Forecasting Approach Combining DBN-Based Deep Neural Networks," Conference Name Unspecified (Your source did not include page numbers or year).

[13] S. K. Jha, C. L. Dewangan and N. K. Verma, "Multi-Step Load Demand Forecasting Using Neural Network," 2019 20th International Conference on Intelligent System Application to Power Systems (ISAP), 2019, pp. 1–6, doi: 10.1109/ISAP48318.2019.9065953.

[14] S. Katruksa and S. Jiriwibhakorn, "Electricity Load Forecasting Based on a Geographic Information System," 2019 5th International Conference on Engineering, Applied Sciences and Technology (ICEAST), 2019, pp. 1–4, doi: 10.1109/ICEAST.2019.8802591.

[15] S. Khan, Z. A. Khan, Z. Noshad, S. Javaid and N. Javaid, "Short Term Load and Price Forecasting using Tuned Parameters for K-Nearest Neighbors," 2019 Sixth HCT Information Technology Trends (ITT), 2019, pp. 89–93, doi: 10.1109/ITT48889.2019.9075062.

[16] W. Kuo, T. Hsieh, H. Chen, C. Chi and Y. Huang, "A Novel Framework for Short-Term Load Forecasting in Micro-grid Energy Management System," 2018 IEEE International Conference on Smart Energy Grid Engineering (SEGE), 2018, pp. 279–283, doi: 10.1109/SEGE.2018.8499492.