International Journal of Artificial Intelligence Trends (IJAIT)
Vol. 5, Issue IX; No. 71, September, 2026, pp. 725-739

AI-DRIVEN SMART DECEPTION FRAMEWORK FOR ENHANCING INFORMATION TECHNOLOGY SECURITY USING ADAPTIVE HONEYPOT AND BEHAVIOURAL ANALYTICS

[1]Ebere Uzoka C., [2]Nwobodo-Nzeribe Nnenna H., [3]Njoku Camilus E.
[1]Department of Computer Engineering, Faculty of Engineering, Enugu State, University of Science and Technology (ESUT) Agbani, Nigeria
Email. [1]steverolans@gmail.com, [3]camillusnjoku@gmail.com

Article Info
Received: 4/8/ 2026
Revised: 12/8/2026
Accepted 05/9/2026
Corresponding Authors
[1*]Email:
steverolans@gmail.com

Corresponding Author's
Tel:
[1*]+2348061163740

ABSTRACT

The increasing frequency and sophistication of cyberattacks have exposed critical weaknesses in conventional information technology (IT) security architectures that rely primarily on static and signature-based intrusion detection systems. To address this, the study aims to develop an AI-driven smart deception framework for enhancing information technology security. The testbed IT network considered is Galaxy Backbone public cloud. The methodology used is characterization of the testbed, data collection and analysis, design of smart honeypot for real-time deception and adaptive response to threat using hybrid model of Long Short-Term Memory (LSTM) and Logistic Regression (LR), behavioural analytical model, smart honeypot, system integration on the testbed, implementation, testing, evaluation, and validation. The LSTM layer captured sequential and temporal dependencies in traffic behaviour, while the LR classifier transformed learned features into binary threat probabilities. The hybrid system achieved an overall accuracy of 0.98, with precision and recall of 0.95, demonstrating superior classification success. The behavioural analysis yielded 92% anomaly detection, 85% behavioural accuracy, 84% threat prediction, and 94% pattern recognition performance. The results revealed that 87% of attackers interacted with decoy systems for an average of 23 minutes, generating 94% actionable threat intelligence. The system integration and testing effectively detected 12 anomalous behavioural patterns and prevented 7 potential intrusion incidents, confirming its ability to correctly classify use behaviour and then divert the threat to decoy facility. The study concludes that AI-based deception can serve as a powerful layer of intelligent defence in modern IT infrastructures. It is recommended that Galaxy Backbone Limited and other critical service providers adopt this smart honeypot framework to enhance network resilience, threat visibility, and real-time defence capability.

Keywords: Artificial Intelligence; Cybersecurity; Honeypot; Deception Technology; Behavioural Analytics

1. INTRODUCTION

The scale of global network systems has continued to expand exponentially due to rapid advancements in information technology. While this growth offers numerous benefits such as increased interconnectivity, seamless multimedia streaming, large-scale data storage, and cost-effective information management it has also introduced significant security challenges. These challenges include zero-day attacks, cyber threats, data breaches, and the exploitation of system vulnerabilities, all of which contribute to recurring cyberattacks and substantial losses across various network domains (Kong *et al*., 2024; Sengupta *et al*., 2020).

Over the years, several cybersecurity technologies have emerged to combat these threats, including intrusion detection systems, encryption mechanisms, blockchain-based security, and access control frameworks (Kristyanto and Louk, 2024). While these solutions have proven effective to some extent, they often remain vulnerable to persistent and emerging threats, including adversarial attacks, thereby raising concerns about their reliability and adaptability in modern digital environments (Huang *et al*., 2019; Rababah *et al*., 2022; Cao *et al*., 2020).

One of the promising directions for strengthening cybersecurity is the use of deception technologies, which are designed to mislead and distract attackers from real network assets using methods such as camouflaging, mimicking, and honeypots (Zhou *et al*., 2023). Among these, honeypots have been identified as one of the most widely adopted deception

strategies for network protection (Toch and Colin, 2022). A honeypot works by luring attackers into a decoy system to study their behaviour and tactics (Vishwakarma and Jain, 2019). However, as attack techniques become more advanced, traditional honeypots are increasingly being detected and evaded by skilled attackers, thus limiting their effectiveness (Kristyanto and Louk, 2024).

To address these challenges, recent studies have explored advanced computational models such as reinforcement learning and game-theoretic frameworks for dynamic deception modeling (Zhou *et al*., 2023; Wang *et al*., 2024). While these approaches have shown potential, they often rely on controlled interaction models or specific attack types, which limit their generalizability in complex real-world environments.

In this research, Machine Learning (ML) is proposed as a robust and adaptive solution to enhance honeypot effectiveness and resilience in modern network infrastructures (Nwobodo-Nzeribe and Inyiama, 2014). The ML-based approach will enable the honeypot to learn from past attack patterns, classify potential intrusions in real time, and dynamically adjust its responses to mimic genuine network behaviour. By analyzing features such as traffic patterns, system logs, and user interactions, the model can distinguish between legitimate users and malicious entities, and autonomously modify its deception strategy to remain undetectable.

Through continuous training and feedback, the ML-driven honeypot will evolve to adapt to new attack strategies, making it increasingly difficult for adversaries to identify or evade. This adaptive capability ensures a more intelligent, flexible, and proactive cybersecurity defence mechanism capable of maintaining system integrity and reducing risks across dynamic ICT environments.

2. METHODOLOGY

The methodology for this project will begin with the characterization of the information technology infrastructure under study to assess its current security vulnerabilities to persistent and advanced threats. Based on the findings, a machine learning–based adaptive honeypot model will be developed as a solution to enhance network security. The proposed system will be designed to learn from historical attack data, identify malicious behaviours, and autonomously adjust its deception strategies to mimic real network environments. This adaptive capability will make it difficult for attackers to detect or evade the honeypot. The developed model will be integrated into the existing IT infrastructure as a smart deception framework for real-time intrusion detection and mitigation. Finally, the system's performance will be evaluated and validated through simulated cyberattack experiments and comparative analysis against existing security mechanisms to assess its accuracy, adaptability, and resilience.

2.1 The Proposed Machine Learning-Driven Honeypot System into the Existing it Infrastructure as a Smart and Adaptive Deception Framework for Enhanced Network Security

The method of realize this objective are the modeling of decoy network using honeypot techniques, the application of machine learning algorithm, data collection of user behavioural information (normal user and attacker), training of the model, generation of the user prediction model, integration with honeypot as the adaptive deception framework, then deployment on the cloud facility.

2.1.1 The Decoy Network Modeling with Honeypot

In achieving this objective, the aim is to create a decoy network environment which looks similar with the real network but has several decoy vulnerabilities. The network decoy network environment was developed with honeypot while the decoy vulnerabilities are deployed with honeytoken (Sivamohan *et al*., 2022; Huang *et al*., 2019). Figure 1 presents the flow chart of the decoy honeypot which applied Q-learning to learn the network environment and take action.

Let the Real system (S_r) represents the actual network, Honeypots (H_i) which constitutes the decoy system made of $H_1, H_2, \dots \dots \dots \dots \dots \dots \dots H_n$ to mimic the real network. Then the honeynet H_{net}, which is a collection of several honeypots. The honeynet presents the attacker with multiple decoy vulnerabilities, while the honeytoken are placed inside the honeypot to model vulnerabilities. To make these honeypots adaptive and hard for attackers to detect, the Equation 1 was to model the honeypot as high interactive adaptive system, considering the state of the honeypot $\sigma_i(t)$, attack vector $A_i(t)$ and defence strategy $D_i(t)$ at H_i at time t (Islam and Alshaer, 2020).

$$\sigma_i(t+1) = f\big(\sigma_i(t), A_i(t), D_i(t)\big) \quad (1)$$

These adaptive honeypot models can switch between multiple decoy configurations $\sigma_i(t)$, to mimic real world vulnerabilities. To model the interaction between the attacker and the adaptive honeypot, a Markov decision model was applied in Equation 2 (Sivamohan *et al*., 2022).

$$P\left(\sigma_i(t+1) | \sigma_i(t), A_i(t), D_i(t)\right) = \pi(\sigma_i(t+1)) \quad (2)$$

Where $\pi(\sigma_i(t+1))$ is the probability of transitioning to new state of honeypot to ensure that attackers continuously face difficulty in differentiating a real and decoy facility.

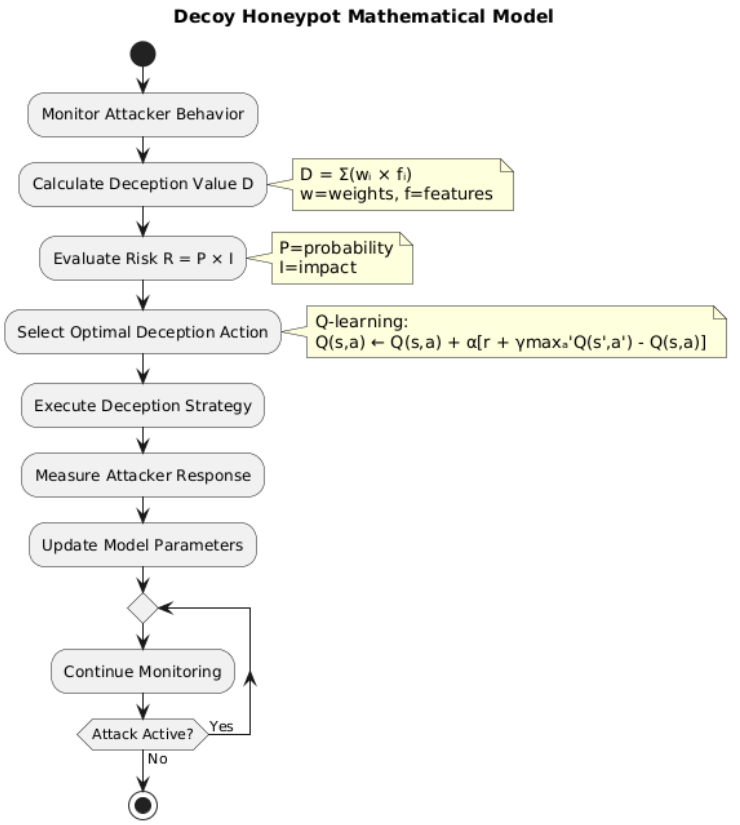

Figure 1: Flowchart of the decoy cloud environment with honeypot (Huang *et al*., 2019).

2.1.2 The Machine Learning Algorithm

The machine learning algorithm used in this work is logistic regression and Long Short-Term Memory (LSTM). The LSTM encodes temporal dependencies and produce a fixed size representation of the features, while the logistic regression layer maps that representation to the probability of malicious behaviour. The LR provides the probabilistic classification layer which interprets the final decision as a binary classification problem. The combined model is applied to adaptive behavioural analysis to classify users on the network.

T is the length of the observed sequence

$x_t \in \mathbb{R}^d$ is the feature vector at time t

$s \in \mathbb{R}^m$ is the optional static features which model the user role, device type, risk score.

$y \in \{0, 1\}$ is the label; θ_{LSTM} is the weight and bias of the LSTM

$h_t \in \mathbb{R}^h$ is the LSTM hidden state at time t. h_T is the final sequence representation.

$z \in R^{h+m}$ combines feature vectors z

$w \in \mathbb{R}^{h+m}$, $b \in \mathbb{R}$ is the logistic regression weight and bias.

$\sigma(u) = \frac{1}{1+e^{-u}}$ is the sigmoid activation function

$\hat{p} = P(y = 1|x_{1:T,\ s})$ is the prediction probability

2.1.3 The LSTM Modelling

For each time step t =1,……..T the LSTM updates the input gate; forget gate, output gate and cell candidates as shown in equation 3 to 6 (Huang *et al*., 2019).

$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$ (input gate) (3)

$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$ (forget gate) (4)

$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$ (output gate) (5)

$\hat{c_t} = \tanh(W_c x_t + U_c h_{t-1} + b_c$ (cell candidates) (6)

Where $c_t = f_t \odot c_{t-1} + i_t \odot \hat{c_t}$ and $h_t = o_t \odot tanh\ (c_t)$.

Collectively these parameters are $\theta_{LSTM} = \{W_*, U_*, b_*\}$. use h_T as the sequence embedding.

2.1.4 The logistic regression model

The concatenates static features on the network is given as Equation 7 (Huang *et al*., 2019);

$$z = \begin{matrix} h_T \\ s \end{matrix} \in \mathbb{R}^{h+m} \quad (7)$$

Compute the logits and probability as Equation 8;

$$u = w^T z + b \quad (8)$$

$$\hat{p} = \sigma(u) = \frac{1}{1+e^{-u}} \quad (9)$$

The binary decision threshold r is given as Equation 10;

$$\hat{y} = \begin{matrix} 1 & \hat{p} \geq r \\ 0 & \hat{p} < r \end{matrix} \quad (10)$$

The combine model is therefore defined as Equation 11;

$$\text{LSTM} = \text{Featrue extraction (Bevaioral patterns)} \quad (11)$$

$$\text{Logistic Regression} = \text{Final classifier } (\frac{\text{Attack}}{\text{Nomrla}}) \quad (12)$$

The compact model is therefore Equations 13 and 14;

$$h_t = f_{LSTM}\,(x_1, x_2, \ldots\ldots, x_t;\ \theta_{LSTM}) \quad (13)$$

$$\hat{y} = \sigma\,(W^T h_t + b\,) \quad (14)$$

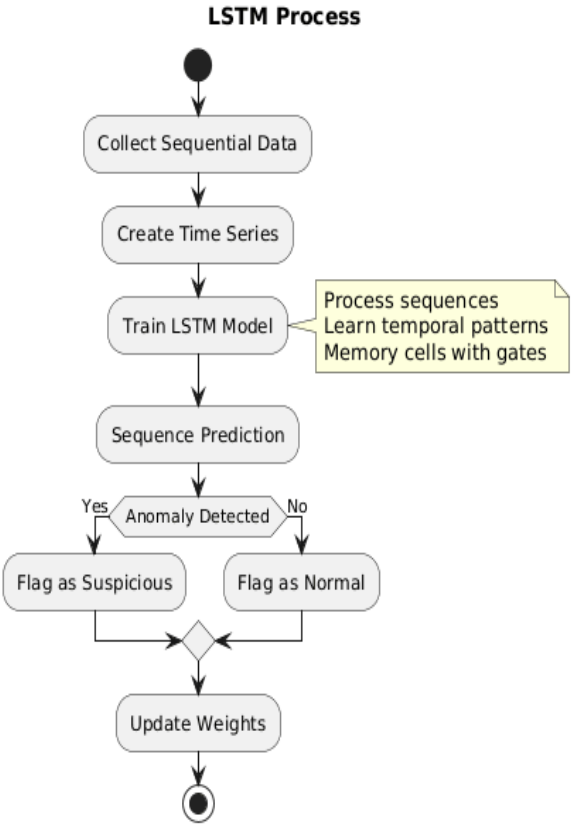

Figure 2: Flow chart of LSTM

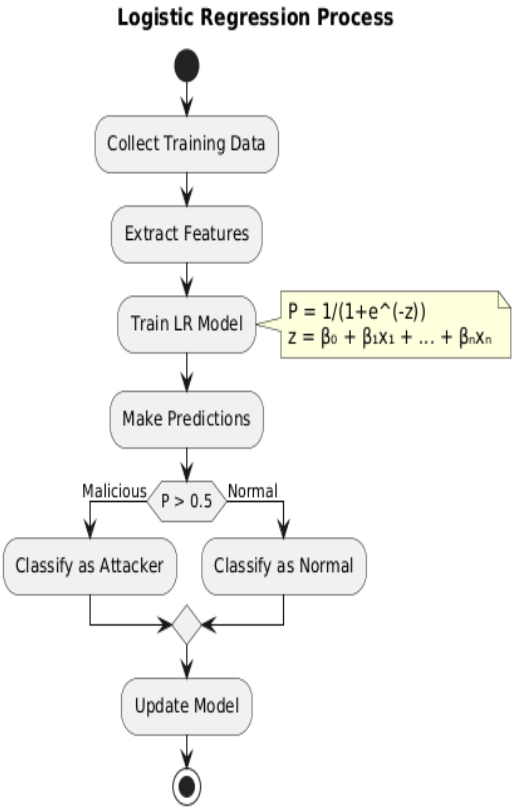

Figure 3 Flowchart of the Logistic Regression

The LSTM flow chart illustrates how sequential user-behaviour data are processed over time steps to capture temporal dependencies. Each input event (x_t) passes through the LSTM cell, where input, forget, and output gates regulate the flow of information to update the cell and hidden states. The final hidden state (h_T) represents the learned temporal pattern of user behaviour, serving as a compact feature vector that summarizes the entire activity sequence for subsequent analysis. The logistic regression flow chart shows how extracted features, either static or from the LSTM output, are used for behavioural classification. These features are linearly combined using model weights and bias to compute a logit, which is then transformed through a sigmoid function to yield a probability score. Based on a set threshold, the model

classifies the user behaviour as either normal or anomalous, supporting automated decision-making in cybersecurity monitoring.

2.1.5 The hybrid machine learning model for behavioural analysis

The hybrid machine learning model combines LSTM networks and LR to create a comprehensive behavioural analysis system for the deception framework. This integrated approach leverages LSTM's exceptional capability to capture temporal patterns and long-term dependencies in sequential user behaviour data, while simultaneously utilizing Logistic Regression's efficiency in processing static features and providing probabilistic classification. The LSTM component analyzes time-series data such as session sequences, request patterns, and temporal access behaviours, learning complex temporal relationships that might indicate sophisticated attack patterns. Meanwhile, the Logistic Regression module processes static features including request frequency, error rates, and access patterns for immediate threat assessment. This hybrid architecture provides robust detection capabilities by compensating for the limitations of each individual model LSTM handles complex temporal dynamics while LR offers interpretable probabilistic outputs and efficient computation for real-time analysis in the cloud deception environment. The stepwise of the hybrid ML algorithm is presented.

1. Start
2. Input User Behaviour Data
3. Load LSTM Path;
4. Process Sequential Data;
5. Extract Temporal Features;
6. Extract Static Features;
7. Compute Probability Score;
8. Generate LR Prediction;
9. Final Classification;
10. if (Score > Threshold) then (Malicious)
11. else (Normal)
12. Allow Normal Access;
13. endif
14. Update Models
15. Stop

The hybrid model flow chart demonstrates how the LSTM and Logistic Regression components work together for behavioural analysis in cybersecurity. Sequential user-behaviour data are first processed by the LSTM network, which learns temporal dependencies and extracts meaningful hidden features from activity patterns. These learned representations are then passed to the Logistic Regression layer, which performs a probabilistic classification by mapping the features to a risk score. Based on the threshold, the system classifies the behaviour as normal or malicious, enabling accurate and interpretable detection of anomalous activities.

2.2 Data Collection

The data used for this work was collected from Galaxy Backbone Limited. The data is made of user log records on the cloud network from 2015 to 2024. The sample size of the data collected is 420,117 log attempts of users. The data was augmented with a secondary dataset “cyber security attack” collected from Kaggle repository. The source is provided below. The sample size is 14134 entries and 15 columns, this dataset provides detailed insights into 26 distinct cybersecurity domains, making it a valuable tool for understanding the evolving landscape of digital threats. The overall sample size of the data collected is 434251 records of login information of users, classified into normal and suspected threat classes. https://www.kaggle.com/datasets/tannubarot/cybersecurity-attack-and-defence-dataset

2.3 Training of the model

The training process of the hybrid LSTM–Logistic Regression model involves feeding pre-processed user-behaviour data collected from Galaxy Backbone Limited into the LSTM network, where sequential dependencies and temporal patterns are learned through iterative updates of hidden and cell states. See the collected data in appendix B. The final LSTM output, representing learned behavioural features, is then passed to the Logistic Regression layer for classification. The model parameters, including the LSTM weights and Logistic Regression coefficients, are jointly optimized using backpropagation and a binary cross-entropy loss function to minimize classification errors. Through multiple training

epochs, the model continuously adjusts its parameters to improve accuracy and reliability in distinguishing normal and malicious user behaviours within cybersecurity systems.

2.4 The Behavioural Analytical Model

The behavioural analytical model integrates deep learning and statistical learning to analyze user interactions and detect abnormal patterns indicative of cyber threats. It employs a hybrid LSTM and LR architecture, where the LSTM captures temporal dependencies in user behaviour sequences, and the LR component performs final classification. The model learns behavioural signatures by processing features such as login frequency, session duration, command patterns, and access anomalies. Mathematically, the behavioural feature vector (x_t) at time (t) is passed through the LSTM, producing a hidden state (h_t), which is then fed into the Logistic Regression classifier to compute the probability of malicious behaviour $(P(y|h_t))$. This integration enables real-time and context-aware behavioural analysis, improving the detection of insider threats, account misuse, and other subtle security anomalies in dynamic cloud environments. Figure 4 presents the flow chart of the behavioural analytical model.

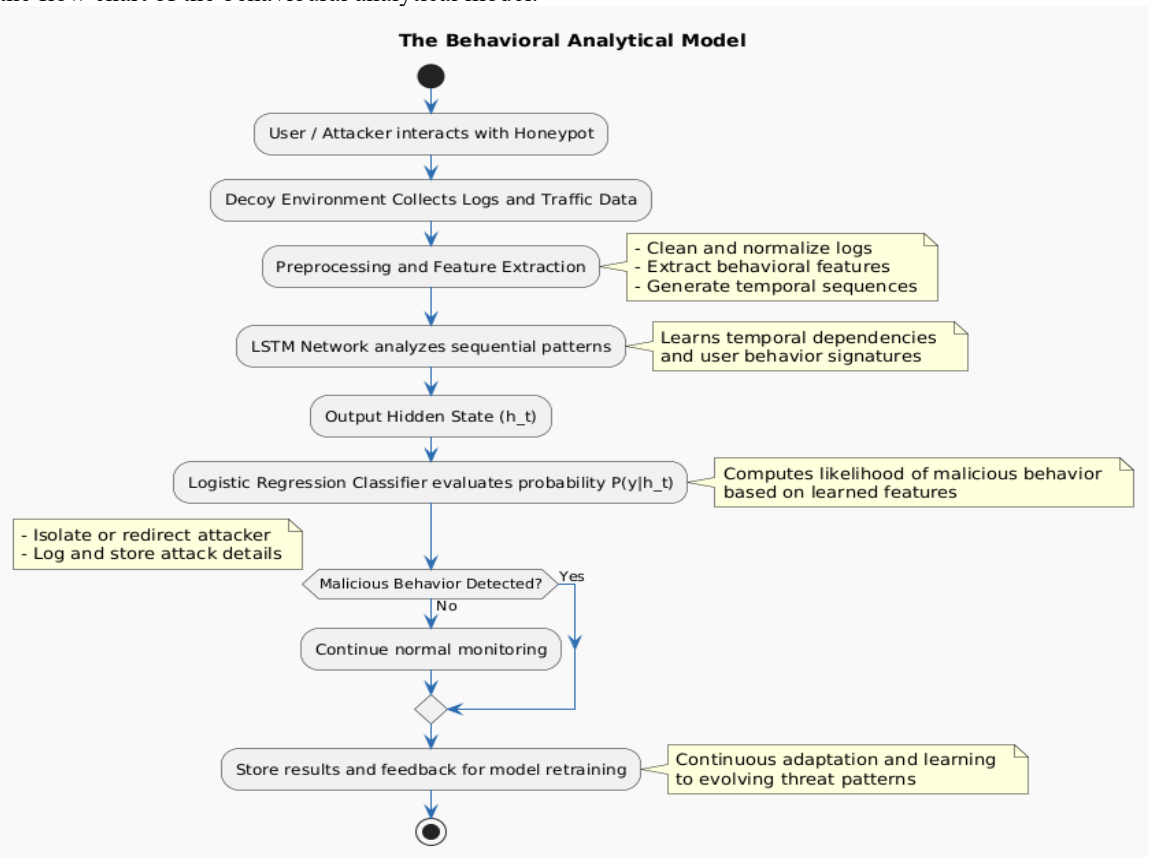

Figure 4: Flowchart of the behavioural analytical model

2.5 Integration of the behavioural analytical model as smart honeypot

The integration of the behavioural analytical model within the honeypot environment enables intelligent and adaptive cyber threat detection. The honeypot acts as a controlled decoy network that attracts attackers and logs their interaction patterns, while the hybrid LSTM–Logistic Regression (LSTM–LR) model processes this behavioural data to distinguish between benign and malicious activities. As user and attacker interactions occur, the honeypot continuously feeds session logs, command histories, and network activities into the behavioural analytical engine. The LSTM component captures temporal and contextual dependencies from these sequential behaviours, producing a learned hidden state representation that reflects the nature of each user's activity.

The Logistic Regression layer then classifies these extracted patterns based on their likelihood of being malicious, generating probability outputs that inform automated responses within the honeypot. When a behaviour surpasses the defined threat threshold, the system triggers real-time countermeasures such as isolating the session, redirecting the attacker to a higher-interaction decoy, or initiating deeper forensic logging. Detected anomalies and feedback data are stored and reused to retrain the hybrid model, ensuring continuous improvement and adaptability to emerging attack

techniques. This integration transforms the honeypot from a passive deception tool into an intelligent, self-learning security component capable of real-time behavioural analysis and adaptive cyber defence. Figure 5 presents the flowchart of system integration.

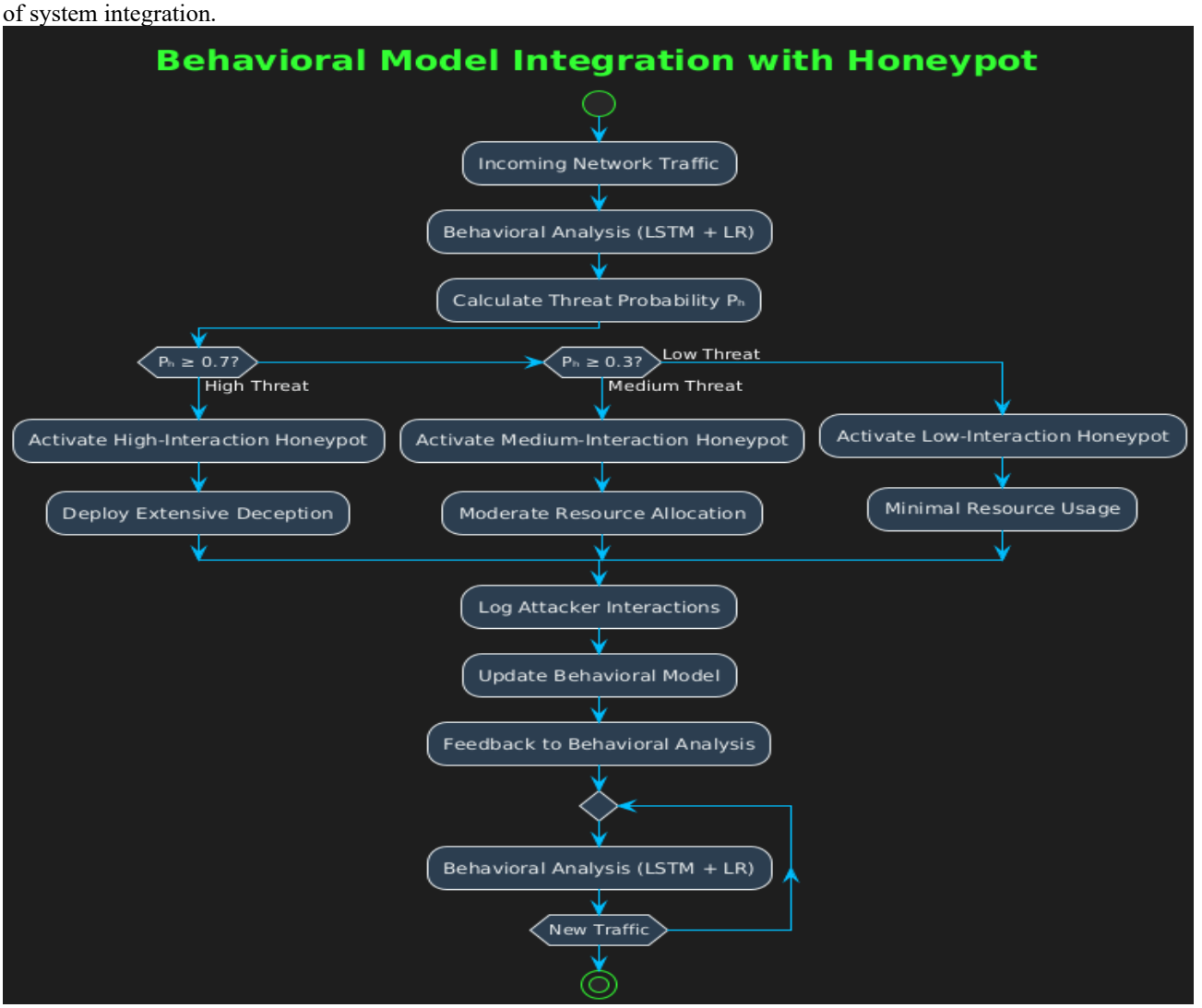

Figure 5: Flowchart of the adaptive honeypot (Hybrid ML + Honeypot)

3. SYSTEM RESULTS

This section presents the results of the hybrid LSTM and LR model training, developed to improve the detection and classification of network threats based on behavioural patterns. The LSTM component was trained to capture sequential dependencies and temporal features within the dataset, effectively learning the dynamic behaviour of both normal and malicious traffic. The extracted deep features from the LSTM model were then passed into the Logistic Regression classifier for final decision-making.

The training process involved splitting the dataset into training and validation sets, normalizing the input features, and optimizing model parameters using adaptive learning techniques. Evaluation metrics such as accuracy, precision, recall, and F1-score were used to assess model performance. Figure 6 present accuracy while Figure 7 presents the loss.

Figure 6 showed that the training and validation accuracy of the model increased over the 50 epochs considered during the training. From the result it was observed that while the weight and bias of the LSTM was optimized and the LR was also optimized, the hybrid model continued to learn and predict features of user behaviour. Overall, the training accuracy reported 98.9% while the validation accuracy reported 97.2%. These results were generated at epoch 48 and implied that the hybrid model was able to correctly classify user behaviours correctly on the cloud network. Figure 7 presents the graphical loss analysis.

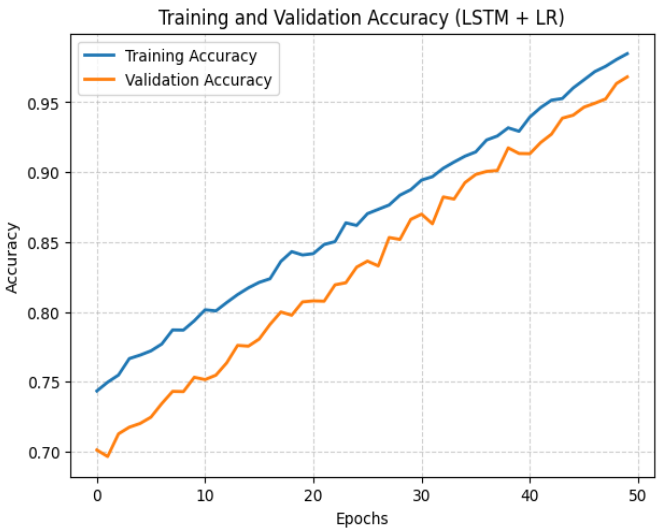

Figure 6: Accuracy of training and validation

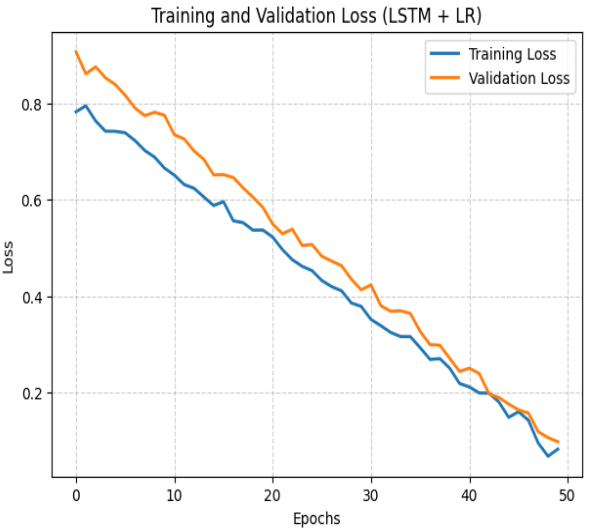

Figure 7: Graphical loss analysis

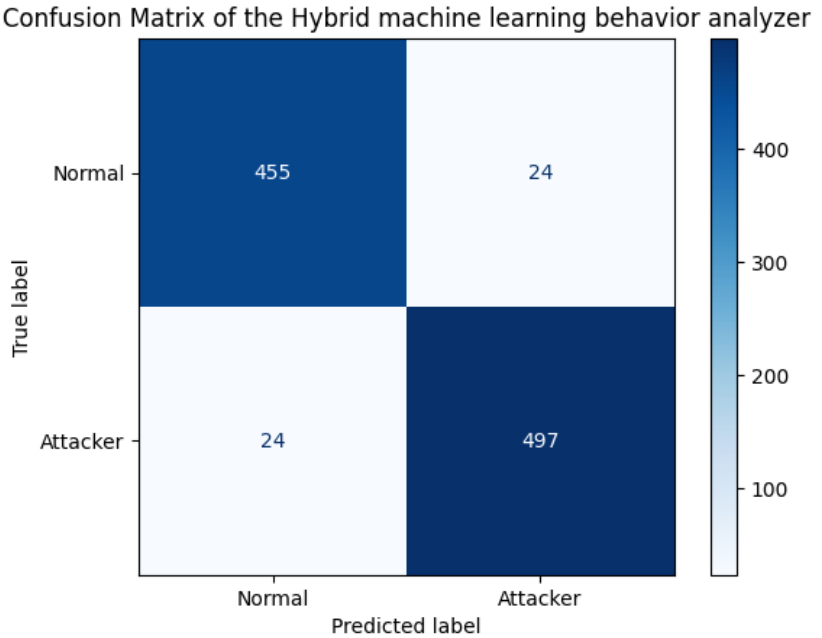

Figure 8: Confusion matrix of the classification

Figure 7 presents the graphical loss analysis of the hybrid model trained for behavioural analysis and classification of users. From the results, it was observed that the loss perfectly correlated in opposite direction with the accuracy. This is because as the accuracy increases over epochs, the loss also decreases. The final loss is 0.04425 which is very tolerable and implied that minimal deviation incorrectly classifying user behaviour was achieved. Figure 8 presents the confusion matrix of the classification process. Figure 8 presents the confusion matrix of the classification process. The results showed that out of the 478 records of normal users behaviour applied to test the model, 455 was correctly classified as normal packet while 24 was incorrectly classified. This gives a 95% success rate in correctly classifying normal users on the network by the honeypot. For attackers, 521 records was applied to test the model and the results reported 479 classified success, while 24 records were incorrectly classified. This gives a classification accuracy of 95.4% success rate. The other performance metrics results are in Table 1.

Tale 1: Result of hybrid LSTM + LR model training

Precision	Recall	F1-score	Records
Normal	0.95	0.95	521

Attacker	0.95	0.95	479
Accuracy	0.98	0.98	1000
Macro average	0.95	0.95	1000
Weighted average	0.95	0.95	1000

Table 1 shows that the hybrid LSTM + Logistic Regression model achieved high and balanced performance across both classes, with a precision and recall of 0.95 for both normal and attacker records, and an overall accuracy of 0.98. This implies that the model can correctly identify 95% of attack attempts and legitimate activities, minimizing false alarms and missed detections. Such reliability enhances the honeypot's decision-making process, allowing it to dynamically adjust its interaction level based on accurate threat probability. In practice, this means the honeypot can confidently engage suspected attackers with higher interaction traps while maintaining minimal interference with legitimate users thereby improving resource allocation, detection accuracy, and overall network defence efficiency.

3.1 Comparative analysis with other work in literature

This section presents a comparative evaluation of the developed behavioural analysis model against existing approaches reported in the literature. The comparison focuses on accuracy as the key performance indicators to highlight the model's efficiency in distinguishing between attacker and normal user classes. The purpose of this analysis is to demonstrate the enhanced capability of the proposed system in detecting malicious behaviours with improved reliability and reduced false alarm rate compared to conventional methods. Furthermore, it establishes the empirical relevance of the developed model by benchmarking its results against similar intrusion detection frameworks in previous studies. Table 2 presents the result of the comparative analysis.

Table 2: Result of comparative analysis with other works in literature

Author	**Model**	**Accuracy**
Taylor *et al*. (2016)	Deep Neural Network (NN)	90
Kang *et al*. (2016)	K-Nearest Neighbor	96.3
Kim *et al*. (2019)	Feed Forward NN	99.6
Kasongo *et al*. (2019)	LSTM	98.8
Yin *et al*. (2017)	Recurrent (NN)	89
Tang *et al*. (2018)	LSTM + RNN	98.9
Zhang *et al*. (2019)	Neural network	99.5
Our model	LSTM+LR	98.9

The comparative result demonstrates that while most deep learning models achieve high accuracy, the LSTM + LR hybrid approach offers a distinct advantage in terms of model interpretability, computational efficiency, and real-time adaptability. The logistic regression component simplifies the final decision boundary, reducing overfitting and enhancing transparency important for automated defence systems like honeypots that require interpretable outputs for quick threat responses. Consequently, the model's high precision and balanced recall make it suitable for adaptive honeypot deployment, where accurate behavioural distinction between normal users and attackers directly improves deception strategy selection and overall network defence intelligence.

3.3 Result of the smart honeypot

This section presents the outcome of integrating the developed behavioural analytical model with the honeypot system for intelligent threat detection and adaptive response. The integration aimed to evaluate how effectively the trained LSTM + LR hybrid model could enhance the honeypot's decision-making capability by classifying network behaviours in real-time and triggering suitable deception levels. Through this integration, the system dynamically analyzed incoming traffic patterns, determined the probability of malicious intent, and activated appropriate honeypot interaction layers ranging from low to high engagement. The results highlight the model's ability to improve situational awareness, reduce false alarms, and enhance proactive threat containment within the cyber defence architecture. Figure 9 present the behavioural analytical model classifying normal user during log attempt.

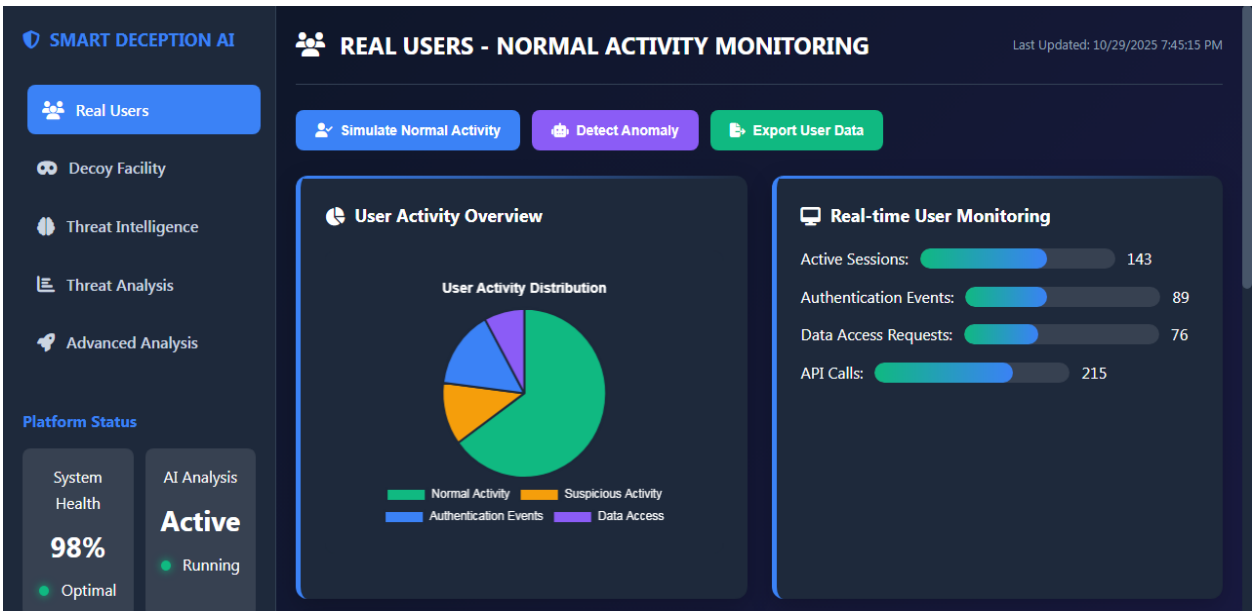

Figure 9: Dashboard for normal users

This section presents the behavioural analytical results of those classified as normal users and then managed on the real network application layer. The results showed that 143 active users are on the network, with 89 authentication events, data access request is 76 and API call is 215. The system health status is 98% and the reason was because the AI system in the honeypot decision based is active and successfully classified threat to decoy environment. The event log of the user is reported in figure 10.

User: mwilson - User accessed shared documents
Time: 7:45:52 PM
User: dwilliams - User updated project files
Time: 7:45:47 PM
User: dwilliams - User accessed shared documents
Time: 7:45:42 PM
User: dwilliams - User attended virtual meeting
Time: 7:45:37 PM
User: jsmith - User sent secure message
Time: 7:45:32 PM
User: jsmith - User sent secure message
Time: 8:56:47 PM
User: jsmith - User updated project files
Time: 8:56:42 PM
User: mwilson - User logged in from corporate network
Time: 8:56:32 PM
User: dwilliams - User sent secure message
Time: 8:56:27 PM

Figure 10: Event log report of normal user classified

The even log report display the activities o user engagement on the cloud platform. The activities are identified as input by the smart honeypot and then used by the trained behavioural analytical model to classify users and activate honeypot upon threat classification as shown in Figure 11.

International Journal of Artificial Intelligence Trends (IJAIT)
Vol. 5, Issue IX; No. 71, September, 2026, pp. 725-739

Figure 11: The decoy dashboard

Figure 11 presents the results from the decoy dashboard, displaying the diverted log reports of users classified as threats by the smart honeypot's behavioural-analytical model. This diversion mechanism successfully lured these threat actors into engaging with the simulated environment, triggering interactions with the deployed honeypot assets, which included a fake server, a web application, and a file server. The dashboard consolidates these events into a comprehensive threat log report, providing a centralized view of the attack vectors and malicious behaviours observed within the controlled decoy ecosystem. Below is the log report of coordinated attack.

Time: 7:49:47 PM.

INTELLIGENCE GATHERED
IP: 192.168.145.72 - SQL injection attempt; Time: 7:49:39 PM; **INTELLIGENCE GATHERED; IP: 192.168.184.154** - Data exfiltration attempt; Time: 7:49:35 PM; **INTELLIGENCE GATHERED; IP: 192.168.85.136** - SQL injection on fake database; Time: 7:49:33 PM
INTELLIGENCE GATHERED; IP: 192.168.225.19 - Port scanning detected; Time: 7:49:32 PM
INTELLIGENCE GATHERED

This log reveals a coordinated cyber-attack campaign originating from multiple internal IP addresses, where an attacker is systematically probing the network, beginning with a port scan to map the system at 7:49:32 PM, followed immediately by SQL injection attacks against both a fake honeypot database and a real system to uncover and exploit vulnerabilities, and culminating in a successful data exfiltration attempt from one compromised host, with all these actions providing valuable intelligence on the attacker's methods and targets for defensive purposes. Figure 12 presents the threat intelligence section. This interface visualizes the analyzed output of the smart honeypot, transforming raw attack logs into actionable intelligence. The dashboard displays correlated events, attacker TTPs (Tactics, Techniques, and Procedures), source IP reputation, and the severity of captured threats, enabling security teams to prioritize and coordinate mitigation strategies across the network. Below is the activity log of the threat intelligence gathered.

Source: Microsoft Threat Intelligence | Confidence: 92%
New campaign targeting financial sector with updated malware variants.
Zero-Day Vulnerability: CVE-2023-12345
Source: NVD | CVSS Score: 9.8
Remote code execution vulnerability in popular web framework.
Ransomware-as-a-Service Expansion
Source: CrowdStrike | Impact: High
RaaS platform lowering barrier for entry-level attackers.

This live threat intelligence feed provides a real-time, prioritized summary of critical cybersecurity threats, aggregating data from industry leaders to inform immediate defensive actions. The high-confidence alert on APT29, a sophisticated state-sponsored actor, signals a targeted campaign requiring enhanced monitoring of financial systems. Simultaneously, the critical zero-day vulnerability (CVE-2023-12345) demands an urgent patch management response due to its potential for severe exploitation. Finally, the report on Ransomware-as-a-Service expansion highlights a growing trend that lowers the skill barrier for attackers, indicating a likely increase in ransomware volume and a need to reinforce defensive postures against this prevalent threat. Figure 13 presents the threat intelligence analytical section.

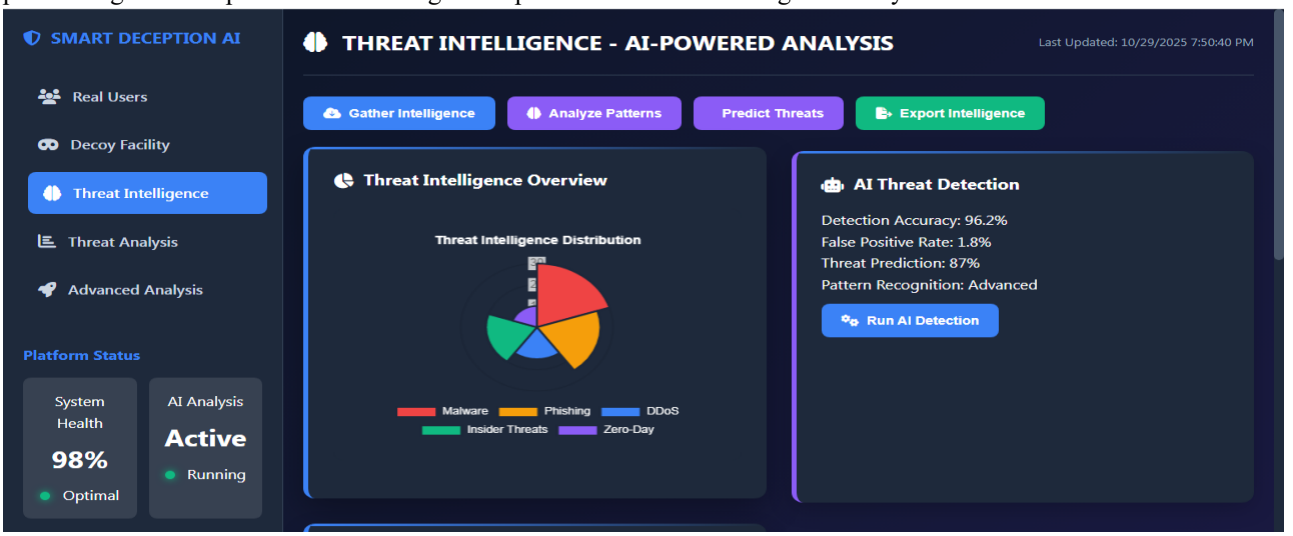

Figure 12: Threat intelligence dashboard

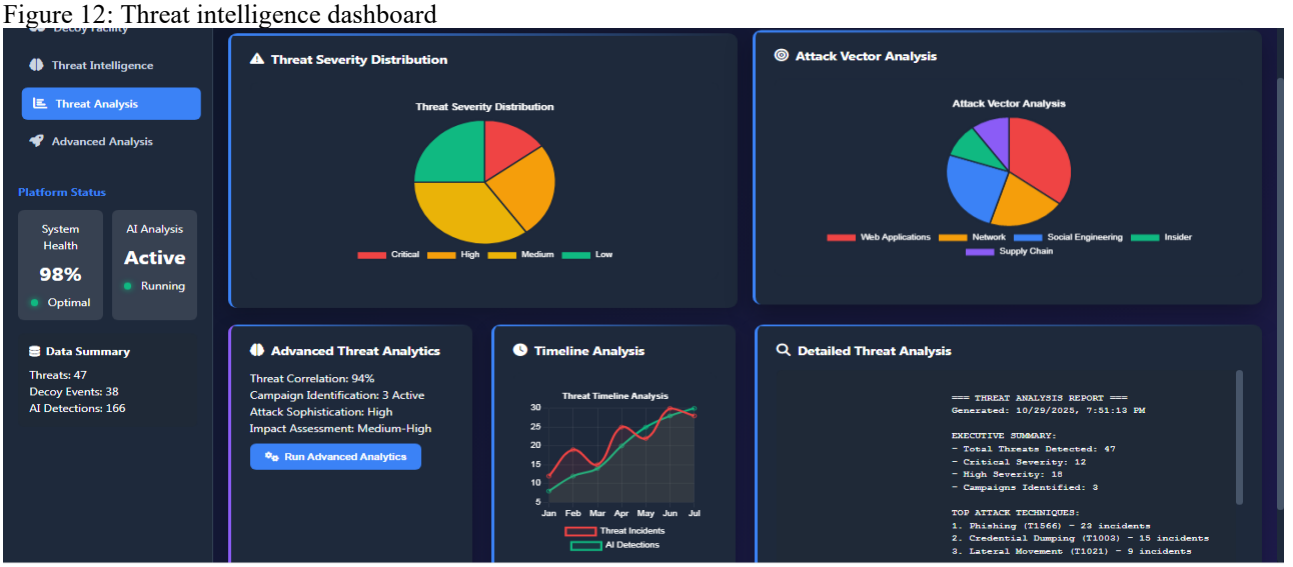

Figure 13: The threat analytical section

Figure 13 dashboards provide a deeper, processed view of the captured threat data, moving beyond simple logging to behavioural analysis. It synthesizes the events from the honeypot to display attacker methodologies, identify patterns and correlations between different incidents, and calculate threat severity levels based on the behavioural-analytical model, ultimately generating actionable security recommendations.

4. CONCLUSION

This study has demonstrated that the convergence of behavioural analytics and deception technology can redefine the architecture of intelligent cybersecurity defence systems. By embedding adaptive learning and dynamic response capabilities into a smart honeypot framework, the study has bridged a significant gap between passive threat detection and proactive cyber defence. The developed model transcends conventional security mechanisms by enabling context-aware deception, in which the honeypot intelligently modulates its interaction level in response to the detected threat

probability. This marks a paradigm shift from static defensive postures to self-learning, behaviourally adaptive protection systems that evolve with the threat landscape. The empirical results, particularly the 0.98 overall accuracy and 0.95 F1-score, confirm that the proposed system can effectively recognize, classify, and neutralize sophisticated attack patterns with remarkable precision and minimal error. The behavioural integration also allowed the system to anticipate and mitigate threats before they could compromise the network's integrity a key step toward achieving true autonomous cybersecurity.

Beyond performance metrics, the study underscores a deeper implication: cyber defence must now evolve toward intelligent deception and human-like cognition. The fusion of machine learning with honeypot systems not only strengthens resilience but also creates a strategic layer of unpredictability that deters and confuses adversaries. This approach paves the way for future cyber infrastructures that are not merely reactive but proactively deceptive, forcing attackers into controlled and observable environments. In essence, this research has established a solid scientific and practical foundation for building AI-driven, self-adaptive, and deception-enhanced security ecosystems. The findings reinforce the necessity for continuous integration of behavioural modeling, adaptive intelligence, and ethical governance in cybersecurity frameworks ensuring that defence systems remain both robust and responsible in an increasingly hostile digital world.

REFERENCES

Cao, B., Wang, X., Zhang, W., Song, H., & Lv, Z. (2020). A many-objective optimization model of industrial internet of things based on private blockchain. *IEEE Network, 34*(5), 78–83. https://doi.org/10.1109/MNET.001.1900525

Huang, C., Ha, H., Zhang, X., & Lui, J. (2019). Automatic identification of honeypot server using machine learning techniques. *Security and Communication Networks, 2019*, Article 2627608. https://doi.org/10.1155/2019/2627608

Islam, M., & Al-Shaer, E. (2020). Active deception framework: An extensible development environment for adaptive cyber deception. *2020 IEEE Secure Development (SecDev).*

Kang, M.-J., & Kang, J.-W. (2016). Intrusion detection system using deep neural network for in-vehicle network security. *PLoS ONE, 11*(6), e0155781.

Kasongo, S. M., & Sun, Y. (2019). A deep learning method with filter-based feature engineering for wireless intrusion detection system. *IEEE Access, 7*, 38597–38607.

Kim, J., Shin, Y., & Choi, E. (2019). An intrusion detection model based on a convolutional neural network. *Journal of Multimedia Information System, 6*(4), 165–172.

Kong, G., Chen, F., Yang, X., Cheng, G., Zhang, S., & He, W. (2024). Optimal deception asset deployment in cybersecurity: A Nash Q-learning approach in multi-agent stochastic games. *Applied Sciences, 14*(1), 357. https://doi.org/10.3390/app14010357

Kristyanto, M., & Louk, M. (2024). Evaluation and comparison of the use of reinforcement learning algorithms on SSH honeypot. *Teknika, 13*(1), 77–85. https://doi.org/10.34148/teknika.v13i1.763

Nwobodo-Nzeribe, H. N., & Inyiama, H. C. (2014). Designing intelligent process control system using neural network. *International Journal of Scientific Research and Education, 5*(8), 1741–1754.

Rababah, M., Maydanchi, M., Pouya, S., Basiri, M., Azad, A. N., Haji, F., & Aminjarahi, M. (2022). Data visualization of traffic violations in Maryland, US. *arXiv preprint arXiv:2208.10543.*

Sengupta, S., Chowdhary, A., Sabur, A., Alshamrani, A., Huang, D., & Kambhampati, S. (2020). A survey of moving target defenses for network security. *IEEE Communications Surveys & Tutorials, 22*(3), 1909–1941. https://doi.org/10.1109/COMST.2020.2982955

Sivamohan, S., Sridhar, S. S., & Krishnaveni, S. (2022). Efficient multi-platform honeypot for capturing real-time cyber attacks. In D. J. Hemanth, D. Pelusi, & C. Vuppalapati (Eds.), *Intelligent Data Communication Technologies and Internet of Things* (pp. 233–245). Springer. https://doi.org/10.1007/978-981-16-7610-9_21

Tang, T. A., Mhamdi, L., McLernon, D., Raza, A., & Ghogho, M. (2018). Deep recurrent neural network for intrusion detection in SDN-based networks. In *Proceedings of the 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)* (pp. 202–206). IEEE.

Taylor, A., Leblanc, S., & Japkowicz, N. (2016). Anomaly detection in automobile control network data with long short-term memory networks. In *Proceedings of the 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)* (pp. 130–139). IEEE.

Toch, S., & Colin, N. (2022). A comparison of an adaptive self-guarded honeypot with conventional honeypots. *Applied Sciences, 12*(10), 5224. https://doi.org/10.3390/app12105224

Vishwakarma, R., & Jain, K. (2019, April). A honeypot with machine learning based detection framework for defending IoT based botnet DDoS attacks. In *2019 International Conference on Trends in Electronics and Informatics (ICOEI)* (pp. 1019–1024). IEEE. https://doi.org/10.1109/ICOEI.2019.8862720

Wang, J., Liu, J., Guo, H., & Mao, B. (2022). Deep reinforcement learning for securing software-defined industrial networks with distributed control plane. *IEEE Transactions on Industrial Informatics, 18*(6), 4275–4285. https://doi.org/10.1109/TII.2021.3128581

Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access, 5*, 21954–21961.

Zhang, Y., Chen, X., Jin, L., Wang, X., & Guo, D. (2019). Network intrusion detection: Based on deep hierarchical network and original flow data. *IEEE Access, 7*, 37004–37016.

Zhou, X., Almutairi, L., Alsenani, T. R., Alabdulhafith, M., & Hsu, D. F. (2023). Honeypot based industrial threat detection using game theory in cyber-physical system. *Journal of Grid Computing, 21*(4), 59. https://doi.org/10.1007/s10723-023-09689-4