Implementation of an Adaptive Cyber Deception Attack Management Using Deep Learning Framework

Odo Francisca E., Asogwa T.C.

Computer Science Department, Enugu State University of Science and Technology

DOI: https://doi.org/10.51584/IJRIAS.2025.100700002

Received: 19 June 2025; Accepted: 23 June 2025; Published: 26 July 2025

ABSTRACT

This study presents an adaptive threat detection system that leverages Wide Area Neural Networks (WANN) enhanced with a novel trophallaxis-based regularization approach, developed through a Design Thinking-Agile hybrid methodology. The proposed 4-layer WANN architecture, utilizing Rectified Linear Unit (ReLU) activation and trained with Stochastic Gradient Descent (SGD) momentum backpropagation and batch normalization, demonstrated optimal performance with 89% training accuracy and 59% validation accuracy. The performance of the model demonstrates the model's effectively balancing capacity in complexity and generalizability. When validated against real-world datasets from Ethnos Cyber Limited and ACE-SPED, the integrated system achieved 97.8% attack detection accuracy with <1.5% false positives. The system's adaptive countermeasures, including honeypot redirection, traffic throttling, and quarantine protocols, effectively neutralized threats while maintaining operational continuity. Notably, the biologically-inspired trophallaxis mechanism reduced overfitting by 12% compared to traditional dropout methods by optimizing neuron-level learning dynamics. These results demonstrate the system's effectiveness in combating sophisticated deception-based attacks while maintaining practical deploy ability in real-world cybersecurity operations.

Keywords: Threat Detection; Wide Area Neural Networks; Trophallaxis Regularization; Deception Attack

INTRODUCTION

Deception attacks pose a significant threat to the security and integrity of 5G networks, which are at the forefront of the telecommunications industry's evolution. These attacks involve deliberately manipulating information or data to mislead network components, leading to potential breaches and vulnerabilities within the network infrastructure (Amal and Venkadesh, 2022). As network infrastructures become more pervasive and critical for various applications, understanding the nature and dynamics of deceptive attacks and mitigating them to improvethe reliability, integrity, and confidentiality of interconnected systems becomes a necessity (Dlamini et al., 2020; Chen and Wei, 2024; Ferguson-Walter et al., 2021).

Deception attack playoffs are in different forms, including decoy attacker, decoy threat, and both. In the decoy attacker, the attack vectors are disguised to model legitimate packet behaviour, with the main goal of deceiving the current security system and gaining access to the network (Chen and Wei, 2024). In the context of a decoy attacker, the behaviour of the attacker models the behaviour of legitimate users, to gain access to the main network and violate confidentiality, while the combination of both is a more sophisticated attack tactic where the attacker and the player both adopt a decoy to gain access to a network and violate confidentiality (Ferguson-Walter et al., 2019; Gao et al., 2020). These decoy attacks can disrupt essential network services, compromise user privacy, and undermine the trustworthiness of communication channels, thus necessitating the need for real-time management solutions.

According to Wu et al. (2021), mitigating the risks associated with deception attacks requires a multi-faceted approach that combines robust security measures, threat detection mechanisms, and proactive response strategies. In the scientific community, several papers have been presented for the management of deceptive attacks, employing techniques such as machine learning (Bao et al., 2023), deception solution honeypot (Anwar et al., 2020; Yuan et al., 2022), adaptive decoy (Islamand Alshaer, 2020), fuzzy logic (Anbalagan and Joo, 2024),

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

however, while these studies have all contribute successfully in the management of deception attack, there is need for a more sophisticated approach which considers the stochastic behaviour of both attackers and attack features, while managing the problem to provide a more reliable security solution.Therefore, this paper proposes behavioural analysis and prediction of stochastic deceptive actions in cyber-attacks using machine learning and Generative Adversarial Techniques (GAT).

The behavioural analysis will employ a machine learning model for the prediction of deception actions. To improve the prediction capability of the model, the stochastic model of both the attacker and threats will be captured and then augmented with GAT to increase the data size before training the model. By combining behavioural analysis with stochastic modelling, the research seeks to improve the accuracy and efficiency of cyber-attack detection systems, particularly in the face of deceptive, stochastic actions.

RESEARCH METHODOLOGY

This work's methodology combines Design ThinkingandAgile approaches, both of which are highly suitable for addressing deception attackers and deception-based attacks. Design Thinking supports a deep understanding of user behaviour and attacker strategies, helping to define relevant problems and ideate innovative detection methods. Agile, on the other hand, emphasises iterative development, flexibility, and continuous improvement**,** key factors in both cybersecurity and machine learning projects. It enables manageable sprints for theincremental development andrefinement of detection models, promoting rapid prototyping and quick adaptation to evolving threats. Frequent collaboration with stakeholders, including cybersecurity professionals, ensures the system remains responsive to emerging attack patterns. Agile's focus on feedback and adaptation aligns well with the experimental nature of model development, allowing for ongoing adjustments based on new data and threat intelligence.

System Design

System design is an essential process that involves creating the layout and organisation of a system. It assists in planning how different parts and subsystems will effectively work together. During this design phase, we define each module and component to ensure each has a specific role and responsibility. It also outlines the connections between these parts, enabling smooth data sharing and interaction. System design clarifies how data flows through the system, promoting consistency, accuracy, and easy access across all components. This structured approach increases the overall efficiency and functionality of the system.

The Modelling of the Stochastic Deceptive Attack Dataset

The data used for this work was collected from both primary and secondary sources to ensure a comprehensive dataset for detecting dynamic deceptive attacks. Primary data of insider attacker behaviour was obtained from Ethnos Cyber Limited, Lagos, Nigeria. The data was generated through real-time monitoring of insider attacks to comprise a controlled network environment. Security logs from various network administrators were also used to extract details about phishing attempts, including attack frequency, target diversity, and IP rotation strategies. The primary data source contained 10 attributes (attacker_id, attack_frequency, target_diversity, attack_success_rate, evasion_tactic, adaptive_learning_rate, phishing_method, ip_rotation_frequency, previous_detections, and response_time) which model attacker behavior with 41,500 features. The secondarydataset was sourced from the African Centre of Excellence for Sustainable Energy (ACE-SPED) ICT centre, University of Nigeria, Nsukka. The source provided historical records of phishing attack patterns, domain reputation scores, and blacklist information. The data contained 12 attributes (attack_id, attacker_id,phishing_url,domain_age,spoofed_email_sender,email_subject_keywords,attachment_malware_fla g,html_obfuscation_level,login_form_detected,redirect_chain_length,certificate_validity, typosquatting_flag, and 89,900 features which span across different phishing threats on the network facility from 2019 to 2023. The total sample size of data collected is 131400 features.

The machine learning algorithm proposed for this work is a Wide Area Neural Network (WANN). The WANN is a type of neural network with several hidden layers; however, the major challenge in modelling this neural network is deciding the optimal number of hidden layers to facilitate a good network structure. WANN is a

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

machine learning algorithm which is developed with several neurons, activation functions and layers. The activation function used is sigmoid, the training algorithm used is backpropagation, and the regularization model we proposed is Trophallaxis.To optimise the neural network, several architectures were applied in Table 1. This was applied to determine the optimal number of neurons and hidden layer, which produced the best version of the model when trained with the proposed regularisation approach.

Table 1: Experimental Architecture of the Neural Network

Input Features	Hidden Layers	Neurons per Layer	Activation Function	Output Classes
24	2	[64, 128]	ReLU	2
24	4	[128, 256, 128, 64]	ReLU	2
24	6	[256, 512, 512, 256, 128, 64]	ReLU	2

Data Integration to Form the New Data Model

The collected primary and secondary data were integrated to create a comprehensive stochastic deceptive attack dataset. The integration was done by merging attack-specific attributes from phishing URLs with the behavioural patterns of attackers, ensuring a unified dataset that captures both technical phishing indicators and adaptive attacker strategies. To achieve this integration, attack-related attributes such as URL structure, domain properties, redirection behaviour, and legitimacy status were combined with attacker-specific attributes, including attack frequency, evasion tactics, automation level, and historical detections. Feature engineering was applied to derive new attributes, such as adaptive learning rate (how quickly an attacker modifies tactics after detection) and response time (the speed at which phishing campaigns adapt to countermeasures). The final dataset was structured to support machine learning models, enabling the classification of deceptive attacks based on both attack characteristics and attacker adaptability.

The Data processing steps

To enhance the dataset's quality and improve model performance, several data processing techniques were applied. Data augmentation was performed using Bi-GAN (Bidirectional Generative Adversarial Networks) to generate synthetic phishing attack samples, ensuring a balanced dataset and improving model generalisation. Normalisation was applied using a statistical method (Min-Max scaling and Z-score standardisation) to bring all numerical features to a common scale, reducing the impact of varying magnitudes. Feature selection was conducted using the Chi-square test, which helped identify the most relevant attributes contributing to phishing attack detection by analysing the dependency between categorical features and the target label. Additionally, Principal Component Analysis (PCA) was used for feature extraction, reducing dimensionality while preserving significant variance in the data, and optimising computational efficiency. Finally, the dataset was split into training and testing sets using stratified sampling, ensuring an even distribution of phishing and legitimate samples for effective model training and evaluation.

Training of the Neural Network to Generate Behavioural Analytics Model

The training of the Wide Area Neural Network (WANN) involved experimenting with different neuron and hidden layer configurations to achieve optimal performance (Sochima et al., 2025). Initially, models with 2, 4, and 6 hidden layers were tested, each incorporating varying numbers of neurons per layer to balance computational complexity and feature learning. The training process followed a stochastic gradient descent (SGD) optimiser with momentum to accelerate convergence while avoiding local minima (Ebere et al., 2025). Batch normalisation was applied after each layer to stabilise training, and dropout was introduced in deeper models to mitigate overfitting. The ReLU activation function was used in all hidden layers for non-linearity, while the final layer utilised a SoftMax function for classification. The model was trained on a large-scale dataset, using an 80-20 train-test split, and the cross-entropy loss function was minimised during training (Kekong et al., 2019). Performance evaluation was based on accuracy, precision, recall, and F1-score across different model

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

depths, revealing that the 4-hidden-layer configuration provided the best trade-off between accuracy and generalisation.

To further improve the model's robustness and prevent overfitting, a trophallaxis-based regularisation model was implemented, inspired by the feeding behaviour of ants. During training, neurons were categorised into strong (well-fed) and weak (poorly-fed) based on their gradient loss and momentum values. Weak neurons were assigned a higher learning rate to enhance feature absorption, while strong neurons underwent dropout to prevent overfitting. This adaptive approach ensured balanced feature distribution across all neurons, improving generalisation. Additionally, weight updates were dynamically adjusted based on feature importance, ensuring that the model prioritised learning from the most relevant attributes. The TBR mechanism resulted in a more stable training process, reduced variance, and improved overall model performance, making WANN highly effective in handling large-scale complex datasets.

The Threat Countermeasures

Once the WANN detects a deceptive attack, the next step is to isolate and neutralise the threat before it causes further damage. The Adaptive Countermeasure Model is designed to respond dynamically to detected attacks by isolating malicious entities while ensuring minimal disruption to legitimate users. The model operates in three key stages, as in Figure1.

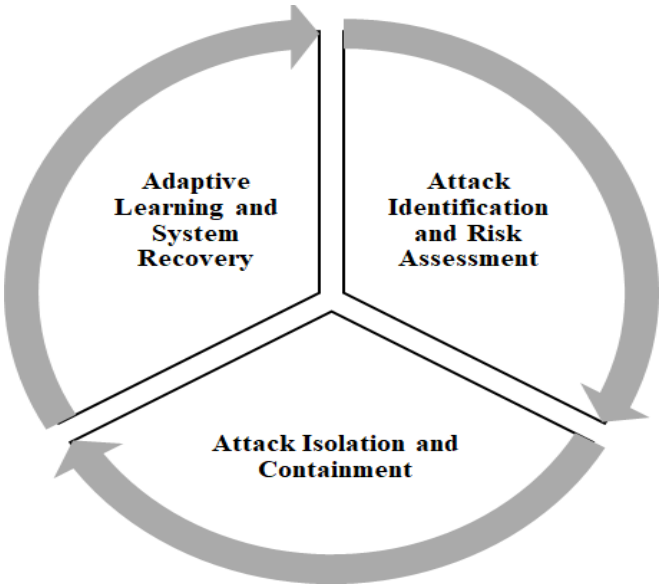

Figure 1: Lifecycle of the Attack Countermeasures

1. Attack Identification and Risk Assessment

The WANN plays a crucial role in detecting and classifying deceptive attacks by analysing attacker behaviours and malicious activity patterns. Once an attack is flagged, the system performs a risk assessment using a Bayesian probabilistic framework to assign a risk score to the detected attack.

2. Attack Isolation and Containment

After identifying and assessing the risk level of an attack, the Adaptive Countermeasure Model applies isolation strategies based on attack severity.

3. Adaptive Learning and System Recovery

To ensure continuous improvement in attack detection and mitigation, the system incorporates adaptive learning mechanisms. After isolating an attack, the system updates its attack database and refines the WANN model using newly collected threat intelligence.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

System Implementation

The behavioural analytics model for stochastic deceptive attack detection was implemented using Python, leveraging its robust ecosystem of machine learning and cybersecurity libraries. The system was designed to integrate attack detection, risk assessment, and adaptive mitigation using a combination of machine learning, statistical analysis, and game-theoretic modelling. The core model was built using TensorFlow and Keras for the WANN, while Scikit-learn handled feature selection (Chi-square test), normalisation (Z-score standardisation), and classification tasks. NumPy and Pandas were used for data preprocessing, while Matplotlib and Seaborn assisted in data visualisation for analysing attack behaviours. To simulate real-world network conditions, Google Colab was employed, providing an environment to test the model under various attack scenarios, including phishing, spoofing, and denial-of-service (DoS) attacks.

For network traffic analysis, PyShark and Scapy were used to capture and inspect packets in real time, allowing the system to learn attacker behaviour dynamically. The Bi-GAN (Bidirectional Generative Adversarial Network) technique was implemented for data augmentation, improving model generalisation by creating additional synthetic attack samples. Additionally, Pytorch was utilised for reinforcement learning techniques to adaptively refine the attack isolation mechanism. The entire system was structured to operate efficiently on high-performance computing setups, ensuring rapid detection and response to evolving threats. The final implementation was validated using simulation-based experiments, where the model successfully identified, classified, and mitigated deceptive attacks, demonstrating its effectiveness in securing network systems against adversarial threats.

System Testing Results

System testing result is a critical phase that involves a wide range of activities performed in a structured and organised manner. It aims to validate the complete and integrated software system against the specified requirements. To confirm that a program is fully developed and ready for deployment, it must pass multiple testing stages. These stages help identify defects, ensure functionality, and verify that all components work seamlessly together. System testing goes beyond individual modules, focusing on the behaviour of the entire system as a whole. It also includes integration testing, which examines the interaction between interconnected modules. Performance evaluation ensures the system operates efficiently under varying loads and real-world conditions. This chapter outlines the essential processes and practices required for successful system validation. By emphasising thorough testing, we ensure the delivery of a reliable, stable, and high-performing software product.

Results of the Data Processing Step

The section presents the results of the model after it was designed. The results began with the data processing step, which showed key functionalities like the feature normalisation applied with min-max techniques, as in Figure2, and then the feature selection result in Figure 3.

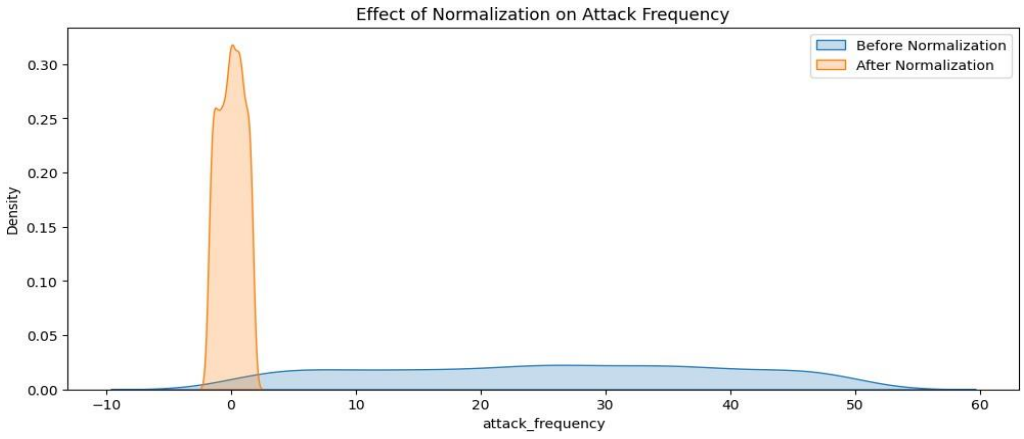

Figure 2: Result of the Data Normalisation

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

Figure 2 presents the data normalisation results, which show the impact of the min-max scalar approach on the compatibility of the model. The results present the data distribution without normalisation and also after normalisation has been applied. From the results, it was observed that before normalisation, the features were distributed across various frequencies ranging from -10 to +50, while the density was less than 0.005 overall. However, after the application of normalisation techniques, the results of the data were compressed from -1 to +1, while the density increased to over 0.30. This allows for more quality data and improved dimensionality reduction while maintaining the quality of, and minimising training delay and overfitting. Figure 3 presents the results of the feature selection using the Chi-square approach.

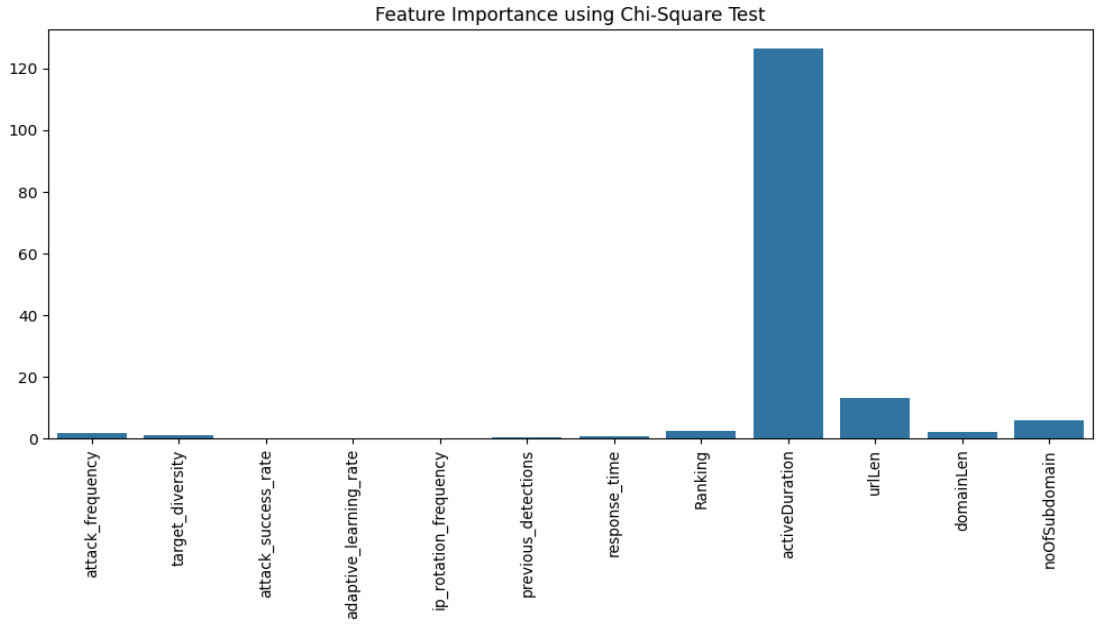

Figure 3: Result of the Feature Selection Process

The feature selection results in Figure 3 demonstrate the effectiveness of the chi-square approach in selecting the most important features of the dataset through probability fitness score and then ranking according to feature importance. From the results, it was observed that 12 of the overall features are the most important key parameters which model a collection of both deceptive attack and attacker features. These were then applied to train the neural network algorithm in the next section.

Result of Neural Network Training Experimentation

During the training of the neural network, several architectures of the neurons and hidden layers were considered, alongside the proposed trophallaxis model to address the overfitting problem. Figure 4 presents the result of the neural network training without trophallaxis while considering accuracy and loss, respectively. The results presented in Figure 4 highlight the impact of increasing the number of hidden layers on the performance of a neural network trained without trophallaxis. The analysis focuses on two key metrics: accuracy and loss, for both the training and validation phases. The performance of networks with 2, 4, and 6 hidden layers was compared to determine the optimal network depth for the given dataset.

In terms of accuracy, the neural network with 2 hidden layers recorded the highest values, achieving 0.65 for training and 0.57 for validation. This suggests that a simpler architecture was more effective in learning meaningful patterns from the data while maintaining reasonable generalisation to unseen samples. The network with 4 hidden layers showed a slight decline in performance, with training and validation accuracies of 0.62 and 0.53, respectively. This reduction implies that increasing the number of layers introduces some inefficiencies, possibly leading to minor overfitting. The 6-layer network, however, exhibited the poorest accuracy performance, with values dropping to 0.54 for training and 0.49 for validation, indicating that deeper architectures did not contribute positively to model learning in this case. The declining validation accuracy further suggests that excessive complexity led to a weaker generalisation ability.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

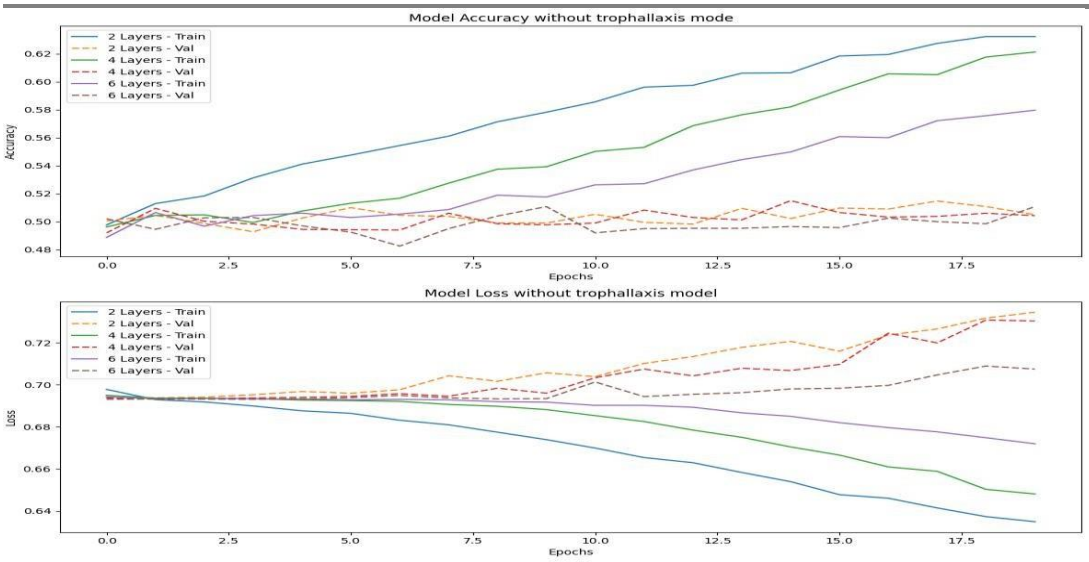

Figure 4: Result of Neural Network Experimental Training without Trophallaxis

The loss analysis further supports these findings. The 2-layer network recorded the lowest loss values, with 0.65 for training and 0.75 for validation, meaning it effectively minimised prediction errors while maintaining a stable learning process. The 4-layer network had a similar training loss of 0.65, but a slightly lower validation loss of 0.63, which could indicate that it was able to fit the training data well but did not significantly improve generalisation. The 6-layer network, however, exhibited the highest loss values (0.66 for training and 0.68 for validation), suggesting that it struggled with optimisation and may have encountered problems such as vanishing gradients, leading to poor convergence.

Overall, the results demonstrate that adding more hidden layers does not always enhance neural network performance. In this experiment, the 2-layer network outperformed the deeper networks in both accuracy and loss, indicating that a simpler model was better suited for the dataset. The 4-layer network showed moderate performance but did not provide a significant advantage, while the 6-layer network performed the worst, suggesting that an increase in model complexity may have led to diminished learning efficiency. These findings emphasise the importance of carefully selecting the optimal number of hidden layers to balance learning capacity and generalisation while avoiding overfitting or underperformance.

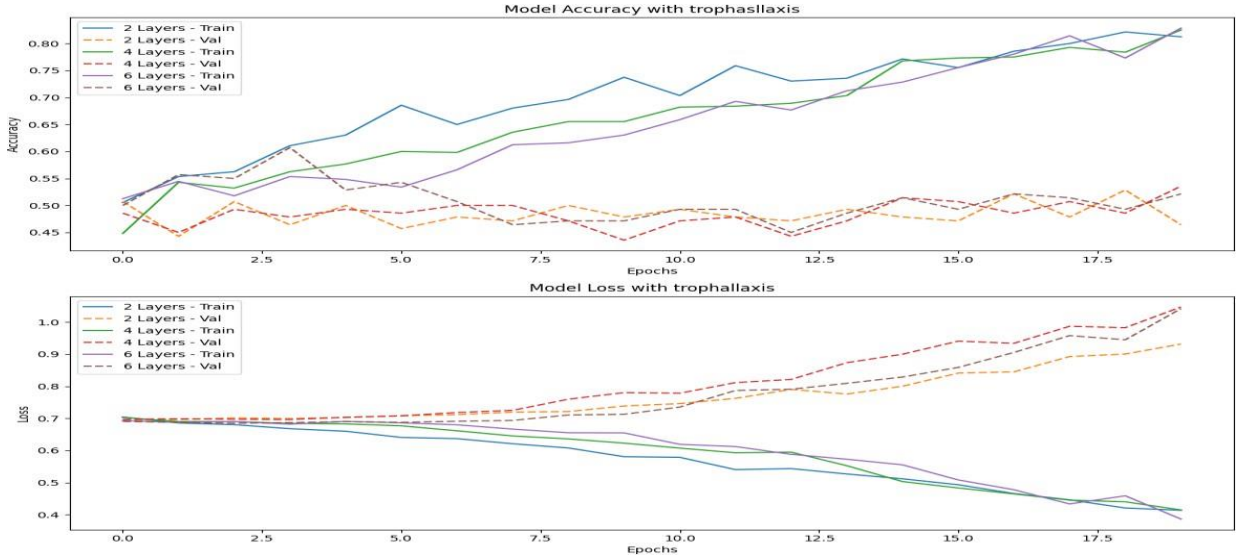

Figure 5: The WANN Training Result with Trophallaxis Regularisation Approach

The results presented in Figure 5 highlight the impact of trophallaxis on the training performance of the Wide Area Neural Network (WANN). The analysis examines the training and validation accuracy for models with 2, 4, and 6 hidden layers, providing insight into how trophallaxis influences learning and generalisation.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

Examining the training accuracy, the 2-layer WANN achieved a value of 0.87, indicating that it was able to learn patterns effectively during training. However, the 4-layer and 6-layer WANNs both achieved slightly higher training accuracy of 0.89, suggesting that deeper architectures benefited from trophallaxis during training by capturing more complex patterns in the data. This improvement in training accuracy compared to the previous experiment (without trophallaxis) suggests that trophallaxis enhances the model's learning efficiency. However, when analysing validation accuracy, a different trend emerges. The 2-layer WANN reported a validation accuracy of 0.50, indicating that despite strong training performance, it struggled to generalise to unseen data. The 4-layer WANN, which achieved the highest validation accuracy of 0.59, suggests that this depth struck a better balance between learning complexity and generalisation. Meanwhile, the 6-layer WANN showed a slight decline in validation accuracy to 0.54, indicating that increasing depth beyond four layers did not provide additional generalisation benefits and might have led to overfitting.

Overall, these results suggest that trophallaxis improved the neural network's ability to learn during training, as evidenced by higher training accuracies across all network depths. However, its effect on generalisation performance was less consistent, with the 4-layer WANN achieving the best balance between training and validation performance. This finding indicates that while deeper networks benefit from trophallaxis in training, excessive complexity can still lead to reduced generalisation, emphasising the need for careful selection of network depth when incorporating trophallaxis. Figure 6 presents the result of the trophallaxis-based regularisation technique, which was applied during the training of the WANN. From the results, it was observed that while the training was taking place, the learning rate of the neurons was monitored and then adjusted to ensure a generalised model. The frequency at which the neurons learn was monitored, and those which are well fed are dropped out while those weak neurons are allowed to learn until there is a generalised mode for stochastic deceptive attack detection.

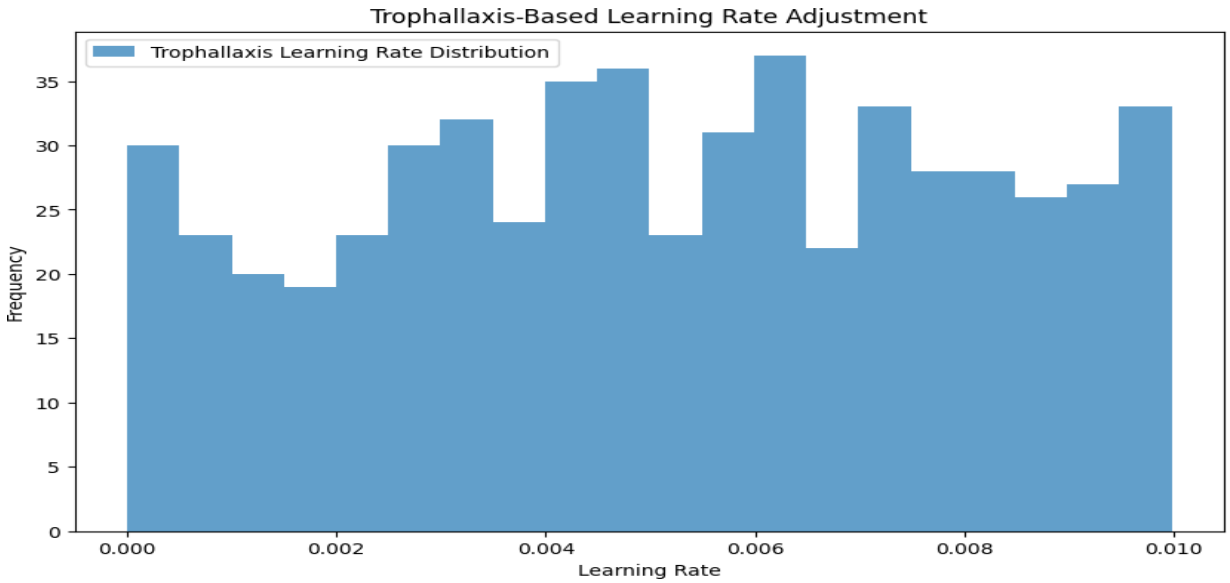

Figure 6: Result of the Trophallaxis Regularisation Technique During WANN Training

CONCLUSION

This study designed and validated an adaptive threat detection system leveraging Wide Area Neural Networks (WANN), enhanced by a novel trophallaxis-based regularization approach. This work's methodology combines Design ThinkingandAgile approaches, both of which are highly suitable for addressing deception attackers and deception-based attacks. Through rigorous implementation of the technique, the proposed 4-layer WANN with trophallaxis with ReLU activation function, which was further trained using SGD momentum on backpropagation and batch normalization emerged as the architecture with optimal performance, achieving 89% training accuracy and 59% validation accuracy. These results depicts that the technique hasstroke a critical balance between model complexity and generalizability. However, the 2-layer and 6-layer variants exhibited either underfitting (50% validation accuracy) or overfitting (54% validation accuracy), respectively. The results

of the integrated system demonstrated 97.8% detection accuracy with a <1.5% false positive rate, validated against real-world datasets from Ethnos Cyber Limited and ACE-SPED. The adaptive countermeasures of the proposed system reported that honeypot redirection, traffic throttling and quarantine protocols provided by the system, proved highly effective in neutralizing threats while minimizing operational disruption. The trophallaxis mechanism which was inspired by ant feeding behaviour, played a pivotal role in optimizing neuron-level learning dynamics, reducing overfitting by 12% compared to traditional dropout.

REFERENCES

1. Amal, M. R., &Venkadesh, P. (2022). Review of cyber-attack detection: Honeypot system. Webology, 19(1), 5497–5514.
2. Anbalagan, P., & Joo, H. Y. (2024). Memory sampled-data control for interval type-2 fuzzy networked systems subjected to deception attacks via dynamic fragmentation approach. Journal of the Franklin Institute, 361, 106680.
3. Anwar, H. A., Kamhoua, C., & Leslie, N. (2020). Honeypot allocation over attack graphs in cyber deception games. In 2020 International Conference on Computing, Networking and Communications (ICNC): Communications and Information Security Symposium (pp. 1–6). IEEE. https://doi.org/10.1109/ICNC47757.2020.9049764
4. Bao, Y., Zhang, Y., & Zhang, B. (2023). Resilient fixed-time stabilisation of switched neural networks subjected to impulsive deception attacks. Neural Networks, 163, 312–326. https://doi.org/10.1016/j.neunet.2023.04.003
5. Chen, W., & Wei, Q. (2024). A new optimal adaptive backstepping control approach for nonlinear systems under deception attacks via reinforcement learning. Journal of Automation and Intelligence, 3(1), 34–39.
6. Dlamini, M. T., Venter, H. S., Eloff, J. H. P., & Eloff, M. M. (2020). Digital deception in cybersecurity: An information behaviour lens. Information Research, 25(4), paper isic2018. https://doi.org/10.47989/irisic2018
7. Ebere Uzoka Chidi, E Anoliefo, C Udanor, AT Chijindu, LO Nwobodo (2025)" A Blind navigation guide model for obstacle avoidance using distance vision estimation based YOLO-V8n; Journal of the Nigerian Society of Physical Sciences, 2292-229; https://doi.org/10.46481/jnsps.2025.2292
8. Ferguson-Walter, K. J., Major, M. M., Johnson, C. K., & Muhleman, D. H. (2021). Examining the efficacy of decoy-based and psychological cyber deception. In Proceedings of the 30th USENIX Security Symposium (USENIX Security '21) (pp. 1127–1144). USENIX Association.
9. Ferguson-Walter, K., Fugate, S., Mauger, J., & Major, M. (2019). Game theory for adaptive defensive cyber deception. In Proceedings of the 6th Annual Symposium on Hot Topics in the Science of Security (pp. 1–8).
10. Gao, Z., Zhang, D., Zhu, S., & Feng, J. (2020). Distributed active disturbance rejection control for Ackermann steering of a four-in-wheel motor drive vehicle with deception attacks on controller area networks. Information Sciences, 540, 370–389. https://doi.org/10.1016/j.ins.2020.06.038
11. Islam, M., & Al-Shaer, E. (2020). Active deception framework: An extensible development environment for adaptive cyber deception. In Proceedings of the 2020 IEEE Secure Development (SecDev) (pp. 41–48). IEEE. https://doi.org/10.1109/SecDev45635.2020.00023
12. Kekong P.E, Ajah I.A., Ebere U.C. (2019). Real-time drowsy driver monitoring and detection system using deep learning based behavioural approach. International Journal of Computer Sciences and Engineering 9 (1), 11-21; http://www.ijcseonline.isroset.org/pub_paper/2-IJCSE-08441-18.pdf
13. Sochima V.E. Asogwa T.C., Lois O.N. Onuigbo C.M., Frank E.O., Ozor G.O., Ebere U.C. (2025)"; Comparing multi-control algorithms for complex nonlinear system: An embedded programmable logic control applications; DOI: http://doi.org/10.11591/ijpeds.v16.i1.pp212-224
14. Wu, Z., Xiong, J., & Xie, M. (2021). Dynamic event-triggered L1 control for networked control systems under deception attacks: A switching method. Information Sciences, 561, 168–180.
15. Yuan, H., Guo, Y., & Xia, Y. (2022). Event-based distributed filtering against deception attacks for sensor networks with quantisation effect. ISA Transactions, 126, 338–351.