Hybrid Deep Learning Enhanced with A Novel Feature Extraction Technique for Detection and Mitigation of Cyber Threats

ASOGWA T.C.[1], EZEH EBERE M.[2]
[1, 2]*Computer Science Department, Enugu State University of Science and Technology*

Abstract- ***This paper proposes a new real-time feature detection and mitigation framework of cyber threats that will be based on hybrid deep learning technique. The system proposed uses Convolutional Neural Networks (CNN) as the part that extracts the spatial features and Autoencoders (AE) as the part that reduces the number of dimensions and detect anomalies resulting in a CNN enhanced with AE architecture that is optimized and can be used in cyber security applications. One of the main novelties of the system is the usage of a feature extraction method based on cross-correlation and it manages to dynamically select the most appropriate network traffic features by assessing the inter-feature relationships over time which provides the model with flexibility in response to a changing trend of threat and discards duplicating or noise data. The system used Python with TensorFlow and Keras to implement deep learning and tested on a virtualized platform with both synthetic. Evaluation terms showed good results, the training accuracy was 89.23%, validation accuracy was 86.74%, and minimal classification loss. The findings have shown that the CNN and AE in combination with feature extraction of the cross-correlation method has great preference to the accuracy and efficiency of network threat detection systems. The framework provides flexible and scalable real-time cybersecurity defence that can be used to reduce false positives and at the same time guarantee prompt mitigating effects.***

Indexed Terms- ***Cybersecurity; Convolutional Neural Network (CNN); Autoencoder (AE); Cross-Correlation Feature Extraction; Real-Time Threat Mitigation***

I. INTRODUCTION

In the last few decades, network technologies have continued to improve the overall quality of services, but at the same time have resulted in increased network security challenges (Elberri et al., 2024). Today, threat vectors such as denial of service, malicious insider threats, man in the middle attack, phishing attack, are some of the few examples of popular attack types, employed by network intruders to unlawfully attack network environments, and violate elements of computer network security which are integrity, availability and confidentiality of data, thus necessitating the need for feature assessment model which investigates the characteristics of packet features towards network environments and classify threats (Ren et al., 2020).

Traditionally, several solutions such are anti-virus, intrusion detection systems, firewalls, etc., are been employed to address issues of threats (Ahmed et al., 2022). However, the sophisticated nature of these threat vectors and their unpredictable nature make these existing solutions not fit to provide the reliable security requirements needed to restore user confidentiality and network integrity (Abdulrahman et al., 2023). Advances in technologies have both positive and negative outcomes. One of the negatives is its adoption by threat attackers to optimize threat features and generate novel threat vectors, which are very difficult to detect using current security systems, and it has remained a research hotspot in the cyber scientific community (Celebi et al., 2023).

Feature assessment is a recent approach that investigates the features of data packets to classify threats in real-time. This approach, due to its effectiveness in distinguishing features of potential threats from normal legitimate features, has recently

dominated studies through approaches such as game theory, machine learning, fuzzy logic, encryption, among other optimization methods (Kim et al., 2020; Parra et al., 2019; Sochima et al., 2025; Li et al., 2020; Rouamel et al., 2022; Ren et al., 2020; Piazza, 2020; Pawlick et al., 2019; Ferguson-Walter et al., 2019). Among these approaches, Machine Learning (ML) is singled out as the most efficient, due to its ability to learn from data directly, model the problem, and then use the reference knowledge for feature assessment to classify the threat.

Some of the papers that applied ML for network feature assessment are Alkhalidi and Yaseen (2021), who applied semi semi-supervised ML technique. Ghosh et al. (2019) experimented with Support Vector Machine (S-VM), Neural Network, Bayes Classifier, and Decision Tree, respectively, to generate feature assessment models and comparatively analyzed threat detection. However, Elberri et al. (2024) revealed that while ML can correctly classify features of packets to identify threats, deep learning provides an even better solution than ML algorithms.

Deep learning (DL) is a type of convolutional neural network with several layers, which has many advantages such as improved accuracy, automated feature extraction capability, more robust feature selection capability, etc (Ebere et al., 2025; Chidi et al., 2024). over ML. Studies such as (Sun et al., 2022; Almousa et al., 2022; Sharma et al., 2022) have all applied DL to manage cybersecurity challenges and recorded good performances; however, (Guo et al., 2023; Elberri et al., 2024) revealed that DL is prone to the problem of over-fitting and false positive results, thus necessitating the need for a real-time cyber threat feature assessment and mitigation framework using cross-correlated deep learning techniques. The cross-correlated approach is tailored towards quality assurance in the feature extraction process to help identify intricate features of cyber threats and legitimate packets, then fed to an improved deep learning algorithm, which will utilize an encoded convolutional neural network for feature identification, concatenation, and classification of threats in real time.

II. RESEARCH METHODOLOGY

The methodology used for this work is the feature-driven development approach. The approach is the best methodology for developing a deep learning feature assessment model for threat mitigation because it emphasizes a user-centered, iterative process that is ideal for addressing complex cybersecurity challenges. By focusing on understanding users' needs and the dynamic nature of threats, this approach enables the development of more practical and adaptive solutions. The iterative nature of design thinking allows for continuous testing and refinement, ensuring that the deep learning models are both effective in detecting threats and flexible enough to mitigate new or evolving risks. This makes it particularly suited for an area where threats are constantly changing, and user interaction is critical for real-world applications.

2.1 Data Collection

The data used in this study was collected from the NSK-DD dataset on Kaggle. NSK-DD provides detailed network traffic records, including normal and attack traffic, making it a reliable source for intrusion detection research. The collected data included 16 features such as source and destination ports, IP addresses, protocol types, packet counts, and statistical flow characteristics. These features were essential in capturing the distinctions between normal and malicious traffic, aiding in the development of an accurate detection model. The sample size of the data collected is 404289 records of network information. This made up the secondary data source, while the primary data source contained similar network attributes, but the test-bed is the National Cyber Security Coordination Center (NCCC), Headquarters, Three Arms Zone, and Abuja, Nigeria. The sample size of data collected is 304333 records of network behaviour, which constitutes both normal and abnormal network information. The total sample size of data collected is 708622, after integration with the secondary data. The data description table is reported in Table 1,

Table 1: Data description

Attribute	Format	Description
SrcPort	Integer	Source port number of the network packet.
DstPort	Integer	The destination port number of the network packet.
SrcIP	Integer	Encoded source IP address of the sender.
DstIP	Integer	Encoded destination IP address of the receiver.
Feature1	Float	Statistical feature related to packet flow timing.
Feature2	Float	Another statistical feature captures network behavior.
Packets	Integer	Total number of packets exchanged in the flow.
Bytes	Integer	Total size of the data transferred in bytes.
Feature3	Float	Computed feature related to packet intervals or delays.
Feature4	Float	Additional statistical metric for traffic behavior.
Value1	Integer	Encoded numerical representation of traffic properties.
Value2	Integer	Another encoded numerical representation of network behavior.
LabelID	Integer	A numerical label representing attack type or normal traffic.
AttackType	String	Classification of network traffic (e.g., DDoS, Normal).
Protocol	String	The transport protocol used (e.g., TCP, UDP, ICMP).
Timestamp	Datetime	The exact time when the network packet was recorded.

2.2 Data Processing

The collected datasets from NSK-DD underwent a structured preprocessing phase to ensure data quality and consistency. The raw data contained noise, redundant entries, and missing values, which were addressed through data cleaning techniques. Feature encoding was applied to make the data compatible with a deep learning model. Following data preparation, an Exploratory Data Analysis (EDA) was conducted to understand the distribution of attack and normal traffic. Statistical summaries and visualization techniques, such as histograms and correlation heatmaps, were used to identify feature relationships and patterns. These analytical steps provided insights into data trends and helped refine the machine learning approach for effective intrusion detection.

2.3 The proposed adaptive online feature extraction approach using the Cross Correlated Extraction (CCE) technique

Cross Correlation Extractor (CCE) is a feature extraction technique that can measure heterogeneous features that are similar, time-varying data, and hence make it suitable as the feature extraction of choice for this work. Traditional CCE, despite its success, may not be able to fully capture the complex dependencies in network packet inflow under varying conditions, thus necessitating the need for an improved adaptive CCE for optimal online feature extraction. The proposed adaptive CCE is made of several components in Figure 1

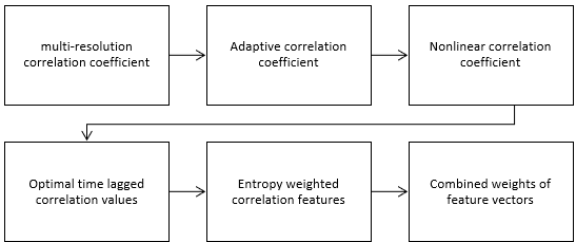

Figure 1: Block diagram of the proposed adaptive CCE

i. Multi-Resolution Correlation Coefficient-

This method computes correlation across different resolutions or scales of network data. It helps in capturing both short-term and long-term dependencies in network features. Multi-resolution techniques, such as wavelet transforms, were used to analyze feature correlations at different levels of granularity, making it useful for detecting anomalies in network traffic at varying time scales.

$$R_{x,y}^{(s)} = \frac{t^{W_x(s,t)} t^{W_y(s,t)}}{\sqrt{t^{W_x^2(s,t)} t^{W_y^2(s,t)}}} \quad (1)$$

Where $W_x(s,t)$ and $W_y(s,t)$ are the wavelet coefficients of the features X (dependent variable) and Y (independent variable) at scale s. $R_{x,y}^{(s)}$ represents the correlation coefficient at specific resolutions.

ii. Adaptive Correlation Coefficient

The adaptive correlation coefficient dynamically adjusts Equation 1 based on the characteristics of the network traffic data. This approach updates correlation weights based on real-time variations, ensuring that the extracted features remain relevant even as network conditions change.

$$R_{x,y}^{(t)} = wt.\frac{\sum_{t=1}^{N}(X_t - X_f)\cdot(y_t - y)\cdot}{\sum_{t=1}^{N}(X_t - X_f)^2 \sum_{t=1}^{N}(y_t - y)^2} \quad (2)$$

Where wt = $\frac{1}{1+e^{-\lambda f_t}}$ at featuring scoring functionf_t, and tuning parameter λ; $R_{x,y}^{(t)}$: This is the corrected correlation coefficient between feature vector x and target variable y, at time step t, adjusted by a weighting function; Xt: The value of the feature x at time step t. yt The value of the target y at time step t; X_f: The mean of the feature X over all time steps t=1 to N; y¯: The mean of the target y over all time steps t=1 to N; The total number of time steps or samples in the dataset; Ft: The feature scoring function at time step t. It quantifies the importance or relevance of the feature X at time t; λ: A tuning parameter that adjusts the steepness or sensitivity of the sigmoid function used in calculating wt; wt: The weighting factor at time t, derived using the sigmoid function wt = $\frac{1}{1+e^{-\lambda f_t}}$. It ensures that features with higher relevance scores have greater influence on the correlation.

iii. Nonlinear Correlation Coefficient

Since network traffic data often exhibits nonlinear dependencies. The nonlinear correlation coefficient, based on a mutual information-based approach, captures complex relationships between features that would otherwise be missed using standard correlation metrics.

iv. Optimal Time-Lagged Correlation Values

Network events often exhibit delayed dependencies; the optimal time-lagged correlation method finds the best time lag for feature relationships, ensuring that delayed effects in network behaviour are effectively captured. The time lag for the features is defined as $R_x, y\,(\tau) = t^{X(t)Y(t+\tau)}$, while the optimal lag $\tau^* = \arg, R_{x,y}\,(\tau)\,\tau$.

v. Entropy Weighted Correlation Features

Entropy is a measure of uncertainty or randomness in data. This approach uses the Shannon entropy techniques to assign weights to correlated features based on their entropy values. Features with high information content (low redundancy) receive higher weights, while redundant and less informative features are given lower importance. This ensures that the most significant network features are prioritized for anomaly detection and intrusion detection models.

vi. Combined Weights of Feature Vectors

This step integrates multiple correlation-based feature extraction techniques by assigning an overall weight to each feature vector. The weights are determined based on a combination of the above techniques, ensuring a balanced feature representation. This approach helps in reducing dimensionality while retaining the most relevant features for cybersecurity analysis.

Algorithm 1: proposed adaptive feature extractor

1. Start
2. Identify packet data from the network
3. Decomposition signal with wavelet and compute the correlation matrix ($R_{x,y}^{(s)}$)
4. Apply $\frac{1}{1+e^{-\lambda f_t}}$ and λ for adaptation of $R_{x,y}^{(s)}$
5. Compute the adaptive cross correlation $R_{x,y}^{(t)}$
6. Apply a kernel function for nonlinear correlation features
7. Determine Optimal time lagged correlation values
8. Apply entropy-weighted correlation features
9. Combine all correlated feature weights
10. Return the final online extracted features
11. End

2.4 Novel Deep Learning Model Using an Encoded Convolutional Neural Network Technique

This section presents the model of the deep learning techniques used for this work. The technique for this work is the CNNand Auto Encoder (AE), then experiments on the different models was also carried out, considering the CNN+AE.

i. Convolutional Neural Networks Model (CNN)

CNNs are potent tools for cybersecurity tasks, particularly in packet analysis. Inspired by the intricate organization of neurons in the visual cortex of animals, CNNs meticulously analyze input images through multiple layers (Lee et al., 2024). By employing convolutional operations with small filters (kernels), they extract vital features such as edges, textures, and shapes. Integrated pooling layers condense spatial dimensions, while fully connected layers facilitate final anomaly classification. During training, CNNs optimize their weights using algorithms like stochastic gradient descent (SGD) and back-propagation, minimizing a loss function that quantifies differences between predicted and actual outputs. This ability to learn hierarchical representations directly from raw data autonomously has revolutionized information security. Figure 2 presents the architecture of the CNN.

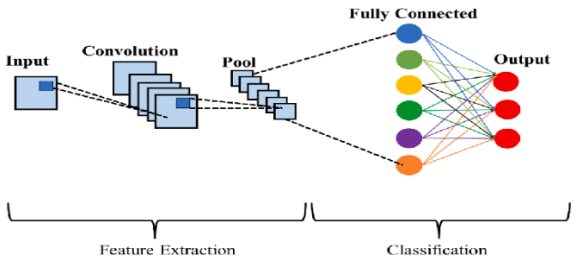

Figure 2: Architecture of the CNN Model

ii. Autoencoder

Autoencoder represents a category of neural networks employed in unsupervised learning tasks, primarily focusing on dimensionality reduction and feature learning. This architecture comprises two main components: an encoder and a decoder. The encoder function compresses the input data into a condensed latent space representation, while the decoder reconstructs the initial input based on this condensed representation. Mathematically, an autoencoder can be represented as follows.

$$Encoder: h = f_\theta(x) \quad (3)$$

$$Decoder: \hat{x} = g_\emptyset(h) \quad (4)$$

For a dataset with n samples, the reconstruction loss L using mean square error (MSE) and mean absolute error (MAE) can be expressed as follows:

$$L_{mse} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x_i})^2 \quad (5)$$

$$L_{mse} = \frac{1}{n}\sum_{i=1}^{n}|x_i - \hat{x_i}| \quad (6)$$

Where x is the input data, h is the latent representation (also called encoding), $\hat{x}$ is the reconstructed output, and f_θ and $g_\emptyset$ are the encoder and decoder functions parameterized by $\emptyset$ θ, respectively.

iii. CNN+AE

This model combines the feature extraction strength of CNN with the efficient data compression and reconstruction ability of an Autoencoder. The CNN component extracts meaningful spatial features from high-dimensional input data, such as network traffic flows or medical images, by applying convolutional layers that capture essential structural information. These extracted features are then passed to an Autoencoder, which compresses them into a lower-dimensional latent representation while preserving critical patterns. The decoder component of the Autoencoder attempts to reconstruct the input, ensuring that only relevant features are retained and noise is eliminated. This model is particularly useful in anomaly detection for cybersecurity applications, where it helps identify deviations in normal network behavior while reducing false positives. Figure 3 presents the flow chart of the CNN+AE.

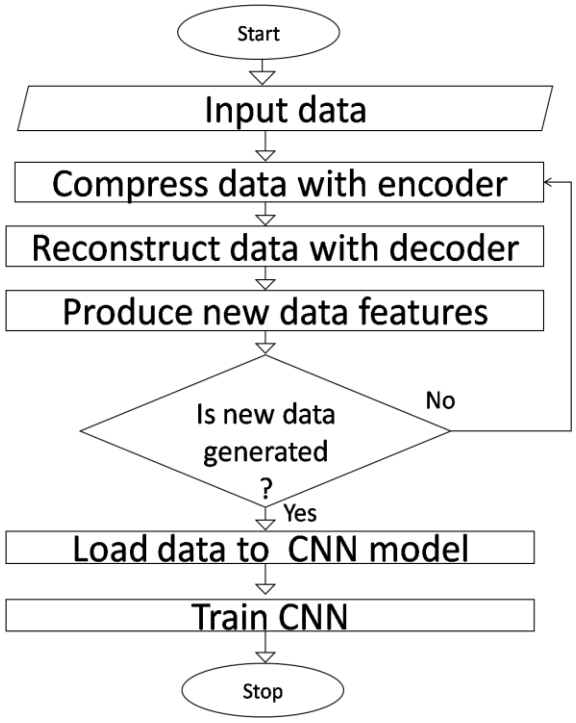

Figure 3: Flowchart of the CNN + AE

2.5 Threat Mitigation

The threat mitigation process in this system begins with the real-time analysis of incoming network traffic using a cross-correlated deep learning model, which is trained to identify complex threat signatures and anomalous patterns. Upon detection of a threat feature, the model triggers a mitigation response. The system then queries user log data, including IP address, session activity, and access timestamps, to establish the context and potential impact of the threat. This correlation allows for precise attribution and rapid decision-making. Concurrently, the transmission control protocol (TCP) stack is invoked to enforce immediate action at the transport layer by terminating the packet stream associated with the threat. The malicious packet is dropped before it reaches the application layer, thereby preventing potential exploitation, data leakage, or lateral movement within the network. This layered response ensures both intelligent detection and swift isolation of harmful traffic, minimizing risk while preserving legitimate network functionality.

2.6 Integration of the Real-Time Cyber Threat Feature Assessment and Mitigation Framework

System integration involved the seamless combination of all core components of the real-time cyber threat feature assessment and mitigation framework into a unified operational environment. The integration process began with the alignment of the deep learning model with the network traffic monitoring engine, ensuring that raw packet data could be pre-processed and fed into the model for real-time threat classification. The model's output was then interfaced with the transmission control protocol layer to enable immediate threat response actions such as packet dropping and session termination. Additionally, the system was connected to a centralized logging and user activity tracking module, allowing for detailed threat context analysis and traceability. API-based communication protocols were used to synchronize various modules, ensuring interoperability. The final integrated system was deployed on a virtualized testbed to validate end-to-end functionality, confirming that detection, analysis, and mitigation operations were executed cohesively and in real time. Figure 4 presents the program flowchart.

The flow chart starts with the identification of incoming packets from the network. This packet is then processed through feature extraction using the cross-correlated approach. These features are then passed to a trained deep learning which performs real-time threat feature assessment. Upon detecting a high-confidence threat, the framework initiates an automated response through the threat mitigation model, which involves accessing relevant system logs, mapping the threat to its source, and executing mitigation strategies such as TCP connection reset, packet dropping, or session isolation. Additionally, a feedback loop should be incorporated to continuously update the model based on new threat patterns, thereby enhancing its adaptability and detection accuracy.

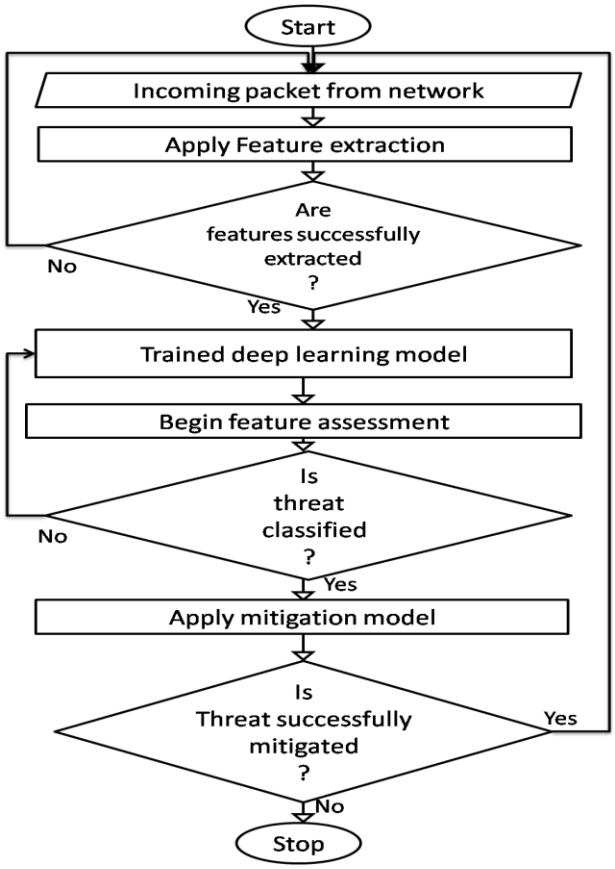

Figure 4: Program flowchart

2.7 System Implementation

The implementation of the CNN+AE-based network assessment model was carried out using the Python programming language. Several libraries were applied to facilitate data processing, model training, and performance evaluation. The primary libraries used include TensorFlow and Keras, which were employed for constructing and training the deep learning models.

NumPy and Pandas were utilized for handling and processing large datasets, while Scikit-learn was used for feature selection, data splitting, and performance evaluation metrics. Additionally, Matplotlib and Seaborn were integrated to visualize the model's results, including accuracy trends, loss curves, and correlation matrices.

The model was trained using an online feature extraction approach, where real-time cyber threat data was processed dynamically. The cross-correlation feature selection method was implemented to identify the most relevant features, ensuring that the model focused on high-impact indicators of cyber threats. The training dataset contained various cyber-attack types, such as DoS (Denial of Service), Ransomware, Botnet, and Man-in-the-Middle (MITM) attacks, with labels indicating normal and malicious activities. The dataset was processed through normalization and encoding to enhance the model's learning capability.

During training, the dataset was split into training (70%), validation (15%), and testing (15%) subsets. The model was trained over 100 epochs with an adaptive learning rate and the Adam optimizer to minimize the categorical cross-entropy loss. Performance evaluation was conducted using multiple metrics, including accuracy, precision, recall, and F1-score.The experimental results were visualized through performance graphs, including accuracy and loss curves, confusion matrices, and ROC curves for each attack category. These visualizations demonstrated the model's effectiveness in classifying different cyber threats with minimal misclassification.

III. SYSTEM TESTING AND RESULTS

This section presents the testing procedures, performance evaluation metrics, and outcomes obtained from implementing the proposed real-time cyber threat feature assessment and mitigation system. System testing was conducted in a controlled network environment using both synthetic and real-world traffic datasets to simulate various cyberattack scenarios, including R2L, U2L, denial-of-service (DoS), and packet injection. The objective was to validate the framework's ability to accurately detect and mitigate threats in real time. Key performance indicators such as detection accuracy, response time, precision, and recall were measured to assess the effectiveness and robustness of the developed model. The results are analyzed to demonstrate the efficiency of the deep learning-based threat detection mechanism, the responsiveness of the mitigation module, and the overall impact on network stability and security.

3.1Result of Feature Extraction

The feature correlation matrix fitness curve over time evaluates the effectiveness of the feature extraction process in selecting the most relevant attributes for network analysis. The fitness score on the y-axis represents the degree of correlation and relevance of extracted features, while the x-axis represents time, showing the progression of feature extraction efficiency. This analysis with the graph in Figure 5 is essential for understanding how well the cross-correlated deep learning approach refines features for improved network security assessment.

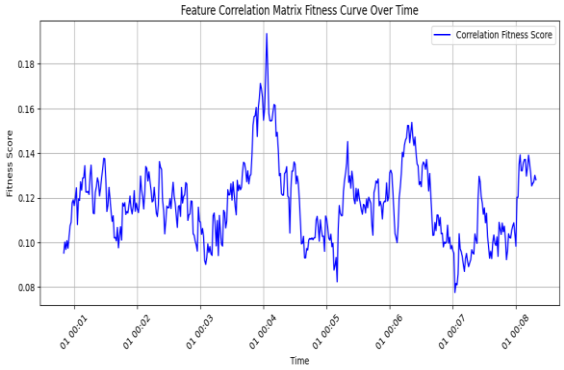

Figure 5: Feature Correlation Matrix Fitness Curve Over Time

From the plot in Figure 5, we observe a fluctuating pattern in the correlation fitness score over time. Initially, the fitness score increases, indicating that the feature selection algorithm is effectively capturing highly correlated attributes in the network data. However, after a brief stabilization phase, a significant peak appears around the 4-second mark, suggesting that the feature extraction model identified a set of features with maximum correlation at that instance. Following the peak, there is a decline, which can be attributed to dynamic network behavior or redundancy filtering in the feature selection process. The system continuously adjusts to new patterns, discarding

irrelevant or redundant features while maintaining a balance in feature selection efficiency.

The overall fitness score remains above 0.08, indicating that the selected features maintain a meaningful level of correlation throughout the process. However, the oscillations in the fitness curve suggest that the model continuously refines its selections, reacting to network traffic variations and attack patterns. The observed fluctuations highlight the adaptability of the model, ensuring that only the most informative features are retained for real-time cyber threat detection.

The results demonstrate that the cross-correlation technique is highly effective in optimizing feature extraction for network security. The peak correlation phases suggest periods where the extracted features are most informative for anomaly detection, while the declines indicate periods of feature redundancy filtering. This ensures that the network model does not rely on stale or irrelevant features, thereby enhancing the overall accuracy of cyber threat detection.

The Feature Correlation Matrix Fitness Curve provides valuable insights into the efficiency of the cross-correlated feature extraction process. The model dynamically refines its feature selection to maintain optimal network analysis performance, ensuring that cyber threats are detected with high precision. The variations in fitness scores confirm that the method successfully adapts to changing network conditions, filtering out irrelevant features while retaining the most significant ones for real-time threat assessment.

3.2 Result of CNN+AE model training

This section presents the training and validation results of the CNN+AE (Convolutional Neural Network + Autoencoder) model for network feature assessment. The model was trained to classify normal and threat-related network features using a combination of convolutional feature extraction and autoencoder-based anomaly detection techniques.

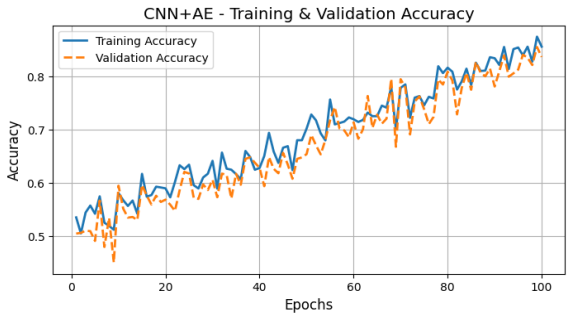

Figure 6: Accuracy of the CNN+AE feature assessment model

Figure 6 illustrates the accuracy performance of the CNN+AE model during training and validation. The results indicate that the training accuracy reached 0.8923, while the validation accuracy was recorded at 0.8674. These values suggest that the model effectively learned feature representations from network traffic data, achieving 89.23% accuracy in training and 86.74% accuracy in validation. The high validation accuracy indicates that the model generalizes well to unseen data, maintaining strong predictive performance. This demonstrates the model's capability in accurately distinguishing between normal network behavior and potential threats using a hybrid approach of convolutional feature extraction and anomaly detection through an autoencoder.

Additionally, Figure 7 presents the loss performance of the CNN+AE model during both training and validation, providing insights into how well the model minimized classification errors.

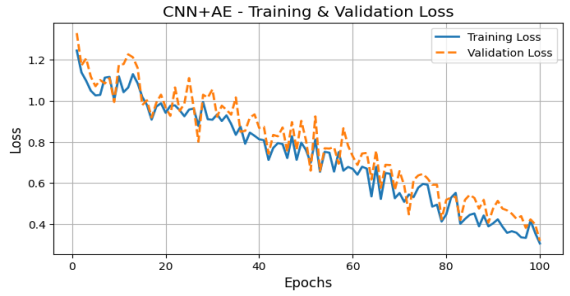

Figure 7: Loss result of the CNN+AE model for network assessment

Figure 7 presents the loss performance of the CNN+AE model during training and validation. The results indicate that the training loss was recorded at

0.3041, while the validation loss was slightly higher at 0.3099. The relatively low loss values suggest that the CNN+AE model effectively minimized errors while learning the distinguishing features of network traffic. The small difference between training and validation loss further confirms that the model generalizes well to unseen data, avoiding overfitting. These results highlight the effectiveness of the CNN-based feature extraction combined with an autoencoder's anomaly detection capability, ensuring robust network threat assessment with minimal classification errors.

CONCLUSION

The proposed and carried out a study later established a new deep learning-based system of real-time cyber threat feature evaluation and mitigation. The central model encourages the usage of Convolutional Neural Networks (CNN) spatial feature extraction and Autoencoders (AE) as a way of dimension reduction and anomaly identification. The essence of the effectiveness of the model is using a novel feature extraction method that makes use of a cross-correlation schema, the feature extraction method dynamically identifies and stores only the most pertinent features on high-dimensional data in the network traffic. It analyzes the real-time network data packets, identifies the important characteristics, based on the correlation approach, and categorizes the information into the CNN+AE model. When the anomalies are detected, the system automatically responds to them by terminating TCP sessions and dropping packets. It was implemented in Python with the use of TensorFlow, Keras and other assisting libraries and deployed into a virtualized testbed environment to test realistic threat scenarios.

The cross-correlation algorithm kept a well-defined cross-correlation index during run time, thus, it consecutively calculated the fitness of retrieved features. This flexibility was found beneficial in that useless, or old features were dropped and only the best explanatory features were employed, and as such, making detection accurate and minimizing false positives. The model attained training accuracy of 89.23, validation accuracy of 86.74 and a small value of losses, which proves the performance. This paper substantiates that feature extraction through cross-correlation plays a crucial role in the development of intelligent, real time- based threat detecting systems. The next steps can be future or the integration of time-based models (such as LSTM), online/continual learning, and even large-scale distributed execution to have an even more flexible and reactive performance.

REFERENCES

[1] Abdulrahman, L. M., Ahmed, S. H., Rashid, Z. N., Jghef, Y. S., Ghazi, T. M., & Jader, U. H. (2023). Web phishing detection using web crawling, cloud infrastructure, and a deep learning framework. *Journal of Applied Science and Technology Trends, 4*(1), 54–71.

[2] Ahmed, N., Ngadi, A. B., Sharif, J. M., Hussain, S., Uddin, M., Rathore, M. S., Iqbal, J., Abdelhaq, M., Alsaqour, R., Ullah, S. S., & others. (2022). Network threat detection using machine/deep learning in SDN-based platforms: A comprehensive analysis of state-of-the-art solutions, discussion, challenges, and future research direction. *Sensors, 22*, 7896. https://doi.org/10.3390/s22207896

[3] Alkhalidi, N., & Yaseen, F. (2021). FDPHI: Fast deep packet header inspection for data traffic classification and management. *International Journal of Intelligent Engineering and Systems, 14*(4). https://doi.org/10.22266/ijies2021.0831.33

[4] Almousa, M., Zhang, T., Sarrafzadeh, A., & Anwar, M. (2022). Phishing website detection: How effective are deep learning-based models and hyperparameter optimization? *Security and Privacy, 5*(6), e256.

[5] Çelebi, M., Özbilen, A., &Yavanoğlu, U. (2023). A comprehensive survey on deep packet inspection for advanced network traffic analysis: Issues and challenges. *NÖHÜ Journal of Engineering Sciences, 12*(1), 001–029. https://doi.org/10.28948/ngmuh.1184020

[6] CHIDI, E. U., UDANOR, C. N., & ANOLIEFO, E. (2024). Exploring the Depths of Visual Understanding: A Comprehensive Review on Real-Time Object of Interest Detection Techniques. Preprints. https://doi.org/10.20944/preprints202402.0583.v1

[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] Elberri, M. A., Tokeşer, Ü., Rahebi, J., Akin, E., & Yilmaz, M. (2024). A cyber defense system against phishing attacks with deep learning game theory and LSTM-CNN with African vulture optimization algorithm (AVOA). *International Journal of Information Security, 23*, 2583–2606. https://doi.org/10.1007/s10207-024-00851-x

[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] Ghosh, A., & Senthilrajan, A. (2019). An approach for detecting spear phishing using deep packet inspection and deep flow inspection. In *Proceedings of the 5th International Conference on Cyber Security & Privacy (ICCS)*.

[11] Guo, Y. (2023). A review of machine learning-based zero-day attack detection: Challenges and future directions. *Computers and Communication, 198*, 175–185.

[12] Kim, J., Camtepe, S., Baek, J., Susilo, W., Pieprzyk, J., & Nepal, S. (2020). P2DPI: Practical and privacy-preserving deep packet inspection. *University of Wollongong, Australia CSIRO Data61, Australia.*

[13] Lee, S., Kareem, A. B., & Hur, J. W. (2024). A comparative study of deep-learning autoencoders (DLAEs) for vibration anomaly detection in manufacturing equipment. *Electronics, 13*, 1700. https://doi.org/10.3390/electronics13091700

[14] Li, C., Guo, Y., & Wang, X. (2022). Towards privacy-preserving dynamic deep packet inspection over outsourced middleboxes. *High-Confidence Computing, 2*, 100033. https://doi.org/10.1016/j.hcc.2021.100033

[15] Parra, G., Rad, P., & Choo, K. (2019). Implementation of deep packet inspection in smart grids and industrial Internet of Things: Challenges and opportunities. *Journal of Network and Computer Applications*, 32–46. https://doi.org/10.1016/j.jnca.2019.02.022

[16] Pawlick, J., Colbert, E., & Zhu, Q. (2019). A game-theoretic taxonomy and survey of defensive deception for cybersecurity and privacy. *ACM Computing Surveys (CSUR), 52*(4), 1–28.

[17] Piazza, N. (2020). *A study on the effectiveness of machine learning techniques to detect and prevent zero-day cyberattacks* [Doctoral dissertation, ProQuest LLC].

[18] Ren, H., Li, H., Liu, D., Xu, G., Cheng, N., & Shen, X. (2020). Privacy-preserving efficient verifiable deep packet inspection for cloud-assisted middlebox. *IEEE Transactions on Cloud Computing*. https://doi.org/10.1109/TCC.2020.2991167

[19] Rouamel, M., Bourahala, F., Guelton, K., Bouzoualegh, S., & Arcese, L. (2022). Fuzzy weighted memory event-triggered control for networked control systems subject to deception attacks. *IFAC PapersOnline, 55*(15), 45–51.

[20] Sharma, S. R., Singh, B., & Kaur, M. (2020). Improving the classification of phishing websites using a hybrid algorithm. *Computational Intelligence, 38*(2), 667–689.

[21] 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

[22] Sun, Y., Chong, N., & Ochiai, H. (2020). Federated phish bowl: LSTM-based decentralized phishing email detection. In *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 20–25).