Digital Innovations & Contemporary Research in SCIENCE, ENGINEERING & TECHNOLOGY
Vol. 13. No. 3, 2025 Series

Article Citation Format
Afolorunso, A.A., Adeniyi, A.E., Abiodun, A.O., Oyewande, O.V. & Garki, F.S. (2025): A Resilient Machine Learning-Driven Multi-Layered Security Model for Modern Cyberspace. Journal of Digital Innovations & Contemporary Research in Science, Engineering & Technology. Vol. 13, No. 3. Pp 27-42
www.isteams.net/digitaljournal
dx.doi.org/10.22624/AIMS/DIGITAL/V13N3P3

Article Progress Time Stamps
Article Type: Research Article
Manuscript Received: 12th July, 2025
Review Type: Blind Peer
Final Acceptance: 11th September, 2025

A Resilient Machine Learning-Driven Multi-Layered Security Model for Modern Cyberspace

[1]Afolorunso, A.A., [2]Adeniyi, A.E., [3]Abiodun, A.O., [4]Oyewande, O.V. and [5]Garki, F.S.
[1]Department of Computer Science, National Open University of Nigeria, Abuja
[2]Department of Computer Science, Bowen University, Iwo, Nigeria
[3]Department of Cybersecurity, National Open University of Nigeria, Abuja
[4and5]Department of Information Systems and Technology, National Open University of Nigeria, Abuja
E-mails: aafolorunsho@noun.edu.ng; abidemi.adeniyi@bowen.edu.ng; aabiodun@noun.edu.ng; ofayemi@noun.edu.ng; fgarki@noun.edu.ng

ABSTRACT

The increasing heterogeneity of cyber threats emphasizes the inadequacy of traditional, single technique security models. Existing models often emphasize perimeter defence, anomaly detection, or access control in isolation, leaving systems vulnerable to advanced consistent threats, attacks from insiders, and data integrity compromise. This escalating advancement of cyber threats, therefore, demands resilient, adaptive, and auditable cybersecurity models that is beyond traditional single-layered defences. This study presents the Multi-Layered Cybersecurity Model (MLCM), which is a hybrid model architecture that integrates three complementary principles: artificial intelligence (AI)-driven intrusion detection systems (IDS), blockchain-based integrity management, and zero-trust access control. Using the Canadian Institute for Cybersecurity Intrusion Detection System 2017 (CICIDS2017) benchmark dataset, a hybrid CNN–BiLSTM intrusion detection model was trained and validated, achieving 98.6% detection accuracy, 98.3% F1-score, Receiver Operation Curve – Area Under the Curve (ROC-AUC) macro of 99.1% and a false positive rate of 1.75%, outperforming standalone CNN (94.8%) and LSTM (95.6%) baselines. The incorporation of blockchain into the model ensured impervious logging with average transaction latency of 0.45 seconds, while Zero-Trust Access policies reduced unauthorized edgewise movements by 84.5% during simulation. The multi-layer feedback loop demonstrated strong flexibility under malicious and incomplete data conditions, maintaining over 96% accuracy despite 9% feature loss. These results support the model's robustness, scalability, and applicability to national cybersecurity strategies, particularly in resource-constrained environments like Nigeria. The MLCM therefore offers a pathway toward strong, flexible, AI-enhanced, and policy-based digital defence in the modern threat landscape as well as advancing resilient digital infrastructures in the era of increasing connectivity.

Keywords: Multi-layered security, Resilience, Blockchain, Zero-trust access, artificial intelligence

1. INTRODUCTION

Cyberspace fosters critical sectors such as education, finance, and even the government. However, its security is increasingly threatened by evolving attack elements. In the current digital age, cybersecurity is a critical enabler of economic, social, and political stability. The recent increased reliance on digital platforms for communication, commerce, and governance has widened the attack surface for malicious actors. Threats such as ransomware, phishing, distributed denial-of-service (DDoS), and insider attacks are escalating in sophistication (Afolorunso et al, 2017; Symantec, 2019; Olalere et al, 2022; Wang et al, 2025). Traditional defense models are most often than not centred around network perimeter and reactive rather than proactive. However, addressing the recent threats requires adaptable and affordable hybrid models that integrate emerging technologies with practical feasibility.

Existing studies underscores the need for multi-layered security models that combine diverse principles into a co-ordinated, adaptive and proactive defense model (Afolorunso and Abass, 2015; Shaukat et al., 2020; Zhang et al., 2022; Soltani et al., 2024). Artificial intelligence (AI) enables anomaly detection beyond signature-based IDS. Blockchain secures integrity through impervious logging and verifiability. Zero-trust architecture (ZTA) enforces continuous and dynamic validation, reducing reliance on static perimeters (Rose et al., 2020; Bensaoud, 2024). However, there has been few studies that integrate these principles into a single, resource-aware model geared for both developed and developing economies. This study proposes an adaptable multi-layered cybersecurity model (MLCM) that combines AI, blockchain, and ZTA to address detection, integrity, and access control challenges pervasively. On their own, these principles are inadequate to secure modern digital systems. There is an urgent need for a consolidated, multi-layered model that combines their strengths while mitigating weaknesses, especially for resource-constrained environments such as Nigeria.

The rest of the paper is organised as follows: Section 2.0 presents the related works, Section 3.0 describes the methodology, Results and discussion of the results is found in Section 4.0 and the paper concludes with Conclusion in Section 5.0.

2. LITERATURE REVIEW

Deep learning has improved intrusion detection by capturing time-related and dimensional traffic patterns that classical approaches often miss (Kim et al., 2016; Soltani et al., 2024). Convolutional Neural Networks (CNNs) and Long Short-Term Memorys (LSTMs) excel in supervised settings but suffer from unbalanced dataset and adversarial maneuver (Goodfellow et al., 2015; Carlini and Wagner, 2017; Mehrban and Ahadian, 2023). Hybrid approaches improve simplism but raise false positive risks (Sharma and Chen, 2020). Interpretability remains a key obstacle to adoption, as analysts require interpretable outputs (Ghosh et al., 2021; Soltani et al., 2024). ZTA operates on the belief of *never trust, always verify* by enforcing continuous authentication, least-privilege policies, and micro-fractionation (Rose et al., 2020; Guo, 2023). Though effective in mitigating insider threats, implementation challenges persevere in cost-sensitive contexts (Moura and Serrão, 2021; Mohamed et al., 2024). Emerging strategies for scalability are modular turnouts and risk-based enforcement. Blockchain ensures data integrity, pedigree, and transparency through unchangeable ledgers (Dorri et al., 2017; Zheng et al., 2018). Approved chains like Hyperledger Fabric balance governance and performance (Androulaki et al., 2018). Nevertheless, blockchain introduces dormancy and storage expenses, requiring optimization through lightweight concord mechanisms (Li et al., 2020).

Hybrid models demonstrate better flexibility and adaptability than isolated principles. Integrating AI with blockchain has enhanced fraud detection and IoT trust management (Salah et al., 2019). Still, comprehensive integration of AI, blockchain, and ZTA remains underexplored, especially in resource-constrained environments. This study is targeted at addressing this gap.

3. METHODOLOGY AND THEORETICAL FORMULATIONS

This study implements and evaluates the Multi-Layered Cybersecurity Model (MLCM) using a realistic simulation approach based on the CICIDS2017 benchmark dataset. Given resource constraints, we simulate an end-to-end implementation where the AI-IDS model is trained and evaluated on selected typical feature statistics from CICIDS2017. The methodology is organized into four parts: Data preparation, AI-IDS model design, Blockchain logging simulation, and ZTA policy enforcement simulation.

3.1 Model Architecture

The MLCM is composed of three layers as shown in figure 1. The first layer is the AI-IDS model which comprises the CNN–BiLSTM hybrid model built to detect anomalies and intrusions. The second layer 2 is the Blockchain Integrity, which is an approved blockchain that store hashed IDS alerts and audit trail records, ensuring verifiability and non-refutation. The third and final layer is the ZTA, which ensures ceaseless verification and least-privilege enforcement to restrict edgewise movement. Cross-layer integration allows IDS detections to trigger blockchain logging and ZTA policy updates, creating a feedback loop that adapts to emerging threats.

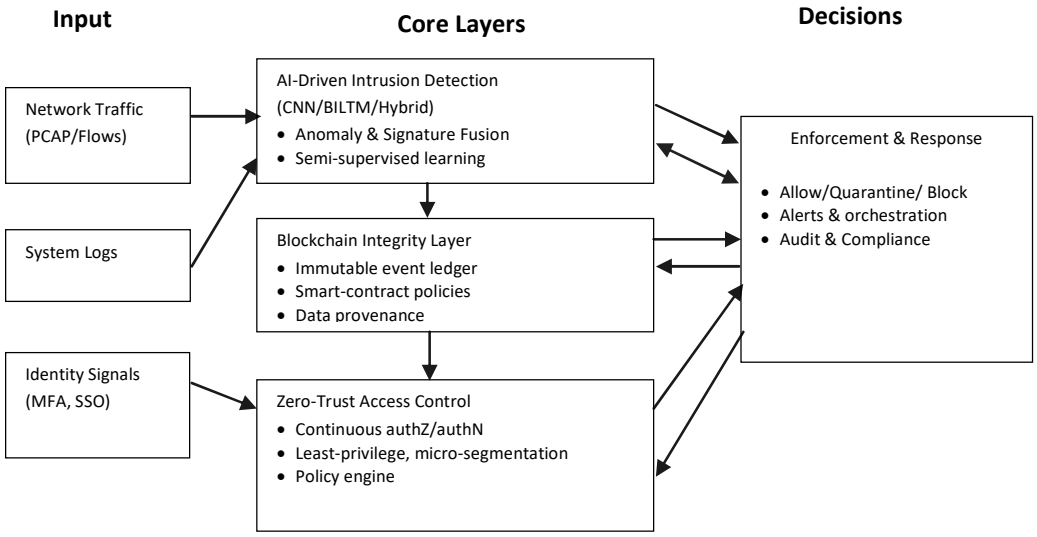


Figure 1: Multi-Layered Cybersecurity Model (MLCM) Architecture.

3.2 Data Preparation

The CNN-BiLSTM intrusion detection model was developed using the full CICIDS2017 benchmark dataset available at http://cicresearch.ca/CICDataset/CIC-IDS-2017/Dataset/CIC-IDS-2017/CSVs/, comprising both benign and malicious network traffic. The CICIDS2017 dataset was selected for its modern attack pattern coverage (Benign, DDoS, PortScan, Botnet, Brute Force, Web Attacks, Infiltration) and realistic network traffic patterns.

Preprocessing activities which include missing-value handling, normalization (Min-Max scaling), one-hot encoding for categorical features (protocol, service), feature selection via mutual information, and class-balancing using SMOTE for minority classes were carried out on the dataset. For the feature set, we concentrate on a compact set of flow-level features such as, flow duration, source/destination bytes, packets per flow, mean packet size, flow inter-arrival times, and protocol flags, which are commonly used in existing works. Non-numeric columns such as IP addresses, Flow ID, and timestamps were excluded.

3.3 AI-IDS Model Design

The AI-IDS architecture follows a hybrid design - a 1D-CNN front-end for local timewise dimensional feature extraction, followed by a Bidirectional Long Short-Term Memory (BiLSTM) to capture sequence dependencies across flows. Input sequences of length 20 are used. The model contains two convolutional layers (kernel sizes of 3 and 5), batch normalization, ReLU activations, a BiLSTM layer with 128 units, and a dense output layer with SoftMax over the intended classes. For the training strategy, we assume the following hyperparameters: Adam optimizer with categorical cross-entropy loss over 20 epochs, learning rate 1e-4, batch size 128, early stopping with patience 5, and class-weighting to mitigate disparity. The theoretical formulation of this algorithm is as given below:

Convolutional Neural Network (CNN):

CNNs, designed to extract **dimensional or local features** from input data like network traffic or intrusion features is given by the equation:

$$Z_{i,j}^{(k)} = f\left(\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X_{i+m,j+n} \cdot W_{m,n}^{(k)} + b^{(k)}\right) \quad \text{..... (1)}$$

Where:

- X is the input feature map,
- $W^{(k)}$ is the weight matrix (kernel) of the k^{th} filter,
- $b^{(k)}$ is the Bias term,
- $f(\cdot)$ is the Activation function (e.g., ReLU), and
- $Z_{i,j}^{(k)}$ is the output feature map after convolution.

This equation defines the convolution operation, which extracts local dimensional patterns from input features using trainable kernels that slide across the input matrix. The ReLU Activation:

The ReLU activation, given by equation (2), introduces irregularity into the model, allowing it to learn complex feature relationships by elliminating negative activations

$$f(x) = \max(0, x) \quad \text{..... (2)}$$

The pooling operation, which reduces the dimensional size of feature maps while retaining the most significant information, improving computational efficiency and generalization, is achieved by Max pooling as given in equation (3)

$$P_{i,j} = \max_{(m,n)\in R(i,j)} Z_{m,n} \quad \text{..... (3)}$$

The fully connected layer is given by:

$$y = f(Wx + b) \quad \text{..... (4)}$$

Where *W* is the weight matrix; *b* is the bias; and *f*(.) is the nonlinear activation ReLU

Long Short-Term Memory (LSTM):
LSTM networks model timewise dependencies and overcome the vanishing gradient problem of traditional Recurrent Neural Networks (RNN)s.

Given input x_t at time *t*, hidden state h_t, and cell state C_t, the forget gate, input gate, cell update and output gate equations are given by equations (5) through (10) below:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \quad \text{..... (5)}$$

where σ is the sigmoid function.

Equation (5) gives the forget gate, which determines which parts of the previous cell state should be discarded, helping the network retain only relevant historical information

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad \text{..... (6)}$$

Equation (6) is the input gate that controls how much new information from the current input will be stored in the cell state. σ is as earlier defined.

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad \text{..... (7)}$$

where tanh is the hyperbolic tangent.

Equation (7) generates candidate cell state values that represent potential new memory content to be added to the network's internal state.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad \text{..... (8)}$$

where * is the element-wise multiplication

The updated cell state combines retained past memory and new candidate values, balancing long-term and short-term dependencies and this given by equation (8).

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad \text{..... (9)}$$

The output gate decides which parts of the cell state contribute to the current output, regulating the information flow to the next layer and is given by equation (9) where σ is as defined before.

$$h_t = O_t * \tanh(C_t) \quad \text{..... (10)}$$

The final hidden state output, given by equation (10), is a gated transformation of the updated cell state, providing context-aware representations for sequence data, and tanh is as defined before.

Bidirectional LSTM (BiLSTM):
BiLSTM processes data in **both forward and backward directions**, capturing past and future dependencies.

The forward LSTM processes the input sequence from past to future, learning timewise dependencies in chronological order and is given by equation (11)

$\overrightarrow{h_t}$ = LSTM$_f\left(x_t, \overrightarrow{h_{t-1}}\right)$ .....(11)

where $\overrightarrow{h_t}$ is the forward hidden state.

The backward LSTM, given by equation (12), processes the sequence in reverse order, capturing contigent dependencies from future inputs.

$\overleftarrow{h_t}$ = LSTM$_b\left(x_t, \overleftarrow{h_{t+1}}\right)$ ..... (12)

where $\overleftarrow{h_t}$ is the backward hidden state.

The forward and backward hidden states are combined by concatenation operation to create a comprehensive representation that includes both past and future context and this is given by equation (13).

$h_t = \left(\overrightarrow{h_t} \oplus \overleftarrow{h_t}\right)$ ..... (13)

where $\oplus$ is the concatenation operation.

Combined CNN–BiLSTM Hybrid Model:
For the hybrid architecture formulated in this paper, **CNN** layers (equation (14)) extracts dimensional (local) patterns representations from the input network traffic data, forming structured feature maps for sequential analysis while **BiLSTM** models timewise relationships in the extracted feature sequences i.e. it processes the CNN-derived features to capture timewise dynamics and long-term dependencies across network flows. This layer is expressed in equation (15). The SoftMax layer, given in equation (16) transforms BiLSTM outputs into normalized probability distributions for multi-class attack categorisation.

$F_{CNN} = f_{conv}(X)$ ..... (14)
h_t = BiLSTM (F_{CNN})..... (15)
$\hat{y}$ = Softmax$(W_h h_t + b_h)$ ..... (16)

Where F_{CNN} is the CNN feature representation, h_t is the hidden representation from BiLSTM, and $\hat{y}$ is the final classification output such as attack category probabilities.

3.4 Blockchain Logging Simulation

To emulate ledger-backed verifiability, IDS alerts are hashed and stored as transactions in a simulated approved blockchain. The simulation models a Hyperledger-style ordering service with features such as average transaction creation cost, block formation every 0.5s, and lightweight consensus. For operational metrics we recorded average logging downtime per transaction, throughput (number of transactions per second – tx/s), and storage overhead relative to raw logs.
The theoretical formulation of this layer is given by equations (17) through (21).

$H(B_t)$ = SHA-256$(B_{t-1} \| T_t \| N_t)$ .....(17)

Equation (17) gives the function that generates a cryptographic hash of the current block B_i by combining the previous block hash B_{i-1}, the transaction set T_i and the nonce N_i. It ensures block immutability and integrity. This hashing equation defines how each block's identity is generated using the previous block hash, the transaction set, and a nonce, ensuring stability and intrusion resistance through cryptographic linkage.

Equation (18) is the chain integrity validation that ensures that each block correctly references its predecessor's hash, preserving the continuous integrity of the blockchain ledger thereby maintaining the chronological continuity and structural security of the blockchain ledger.

B_{i-1}^{hash} = H(B_{i-1}) ⇒ H(B_{i-1}) = prevHash(B_i) .....(18)

There is also the digital signature verification process that authenticates the origin of transactions, confirming that a message M signed with a private key is authentic when verified using the corresponding public key K_{pub} i.e. it verifies a signed message matches the signer's public key, thereby preventing forgery or spoofing. The equation is given by equation (19).
Verify(Sig, M, K_{pub}) =

$$\begin{cases} \text{True}, & \text{if Decrypt}(Sig, K_{pub}) = H(M) \\ \text{False}, & \text{otherwise} \end{cases} \quad \ldots\ldots \quad (19)$$

The consensus-threshold function, given by equation (20), simplifies proof-of-authority i.e. the voting model.

$$C_{valid} = \begin{cases} 1, & \text{if } \frac{\sum_{j=1}^{n} v_j}{n} \geq \theta \\ 0, & \text{otherwise} \end{cases} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (20)$$

Where, v_j represents validator votes, n is the total number of validators, and θ is the consensus threshold. It determines whether a new block is accepted based on a predefined majority or confidence level for instance, 0.67 for two-thirds majority.

Lastly is the blockchain-linked alert verification given by equation (21).

$$A_{verified} = \begin{cases} 1, & \text{if } H(A_t) = H_{chain}(A_t) \\ 0, & \text{otherwise} \end{cases} \quad \ldots\ldots\ldots\ldots \quad (21)$$

This is the part that ensures that intrusion-detection alerts logged on the blockchain are intact and unaltered by comparing stored and computed hashes. It represents how IDS alerts are cross-verified against blockchain records to ensure that no compromise occurred during storage or transmission.

3.5 Zero-Trust Access (ZTA) Formulation

ZTA behaviour was simulated as a policy agent that evaluates contextual signals (user/device posture, recent alerts, geo-location, and time) before granting access. Entities with high recent alert counts or risk-score greater than 0.9 are denied and intermediate risk triggers reauthentication. The simulation measures policy enforcement success rate, additional authentication latency, and false-block rates for legitimate requests when adaptive policies are strict.

There are basically five components (equations) that drives this layer and they are as follows:
The ceaseless authentication function computes a user's trust score at a given time by aggregating identity, device, network, and behavioral context factors. This is given by equation (22)

$$A_{trust}(u, t) = f(C_{id}, C_{dev}, C_{net}, C_{beh}, t) \quad \text{..... (22)}$$

This function computes a dynamic trust score, A_{trust}, for a user u at time t based on several contextual confidence factors such as identity credentials (C_{id}), device attributes (C_{dev}), network context (C_{net}), and last but not the least, behavioural patterns (C_{beh}).

After which comes the trust score normalization given by:

$$T_{norm}(u, t) = \frac{A_{trust}(u,t)}{max(A_{trust})} \quad \text{..... (23)}$$

The normalized trust value T_{norm} guarantees comparability across users and sessions, mapping the trust score to a [0, 1] range for adaptive access decisions. This step basically scales the raw trust score to a standardized range [0,1][0,1][0,1], enabling persistent comparison across sessions and users.

Next comes the risk-adaptive access control given by equation (24).

$$P_{access}(u, r, t) = \begin{cases} 1, & \text{if } T_{norm}(u,t) \geq T_r \\ 0, & \text{otherwise} \end{cases} \quad \text{..... (24)}$$

Here, P_{access} denotes whether a user u is granted access to resource r, at time t, depending on whether their normalized trust score exceeds a risk-based threshold T_r. The risk-adaptive access-control rule dynamically grants or denies access based on whether the current trust score meets or exceeds a resource-specific threshold.

There is also the ceaseless verification and re-evaluation function given by:

$$T_{norm}(u, t + \Delta t) = \propto . T_{norm}(u, t) + (1 - \propto) . f_{new}(u, \Delta t) \text{ ..(25)}$$

Equation (25) models dynamic re-evaluation of trust over time, where f_{new} captures newly observed context or behavior changes, and α controls the weighting between past and current assessments.

It updates a user's trust score over time by blending previous confidence with newly observed context, allowing trust to weaken or strengthen adaptively. The final stage is the enforcement of the *principle of least privilege* by granting the minimal subset of resources *R'* that still allows user *u* to perform legitimate tasks at time *t*. This is depicted by the equation:

$\mathcal{R}_{granted}(u) = \arg\min_{\mathcal{R}' \subseteq \mathcal{R}} \{|\mathcal{R}'| \mid \forall r_i \in \mathcal{R}', P_{access}(u, r_i, t) = 1\}$..... (26)

The least-privilege enforcement equation (equation (26)) ensures that only the minimal necessary set of resources is granted to a user, reducing potential attack surfaces while preserving functionality. Equations (22) through (26) together mathematically represent the core logic of ZTA.

3.6 Evaluation Metrics

For the evaluation metrics, we report:

i) Class-wise precision, which quantifies the percentage of correctly identified positive samples among all positive predictions (i.e. how many predicted positives are correct), indicating reliability and is given by the equation:

Precision = $\frac{TP}{TP+FP}$ ..... (27)
Where *TP* stands for true positive and *FP* stands for false positive

ii) Recall or Sensitivity measures how effectively the model identifies all relevant positive instances, i.e. ability to find all true positives and is given by equation (28).

Recall = $\frac{TP}{TP+FN}$ ..... (28)
Where *TP* is as defined earlier and *FN* stands for false negatives

iii) F1-score provides a balanced measure between Precision and Recall, especially when dealing with class disparity. This is given by equation (29).

Recall = $2\left(\frac{Recall * Precision}{Recall + Precision}\right)$ ..... (29)

For this study, both macro-F1 and weighted-F1 were measured.

iv) The overall accuracy, which is same as overall correctness of the model, measures the proportion of correctly classified instances among all samples, reflecting overall prediction correctness and is given by the equation:

Accuracy = $\frac{TP + TN}{TP + TN + FP + FN}$ ..... (30)
Where *TP*, *FP*, *FN* are as earlier defined and *TN* stands for true negative.

v) AUC(ROC) i.e. Area under the ROC curve evaluates the model's ability to distinguish between classes across different decision thresholds, summarizing its overall prejudicial power. This is given by equation (31).

$$\int_0^1 TPR(FPR)\, d(FPR) \quad \text{..... (31)}$$

Where *TPR* is true positive rate, and *FPR* is false positive rate.

Other operational metrics include support counts, blockchain logging latency (ms), throughput (tx/s), and ZTA enforcement success rate, false-block rate, and added latency. The model was benchmarked against CNN-only and LSTM-only models to assess hybrid learning benefits in intrusion detection.

4. RESULTS AND RESULTS DISCUSSION

4.1 Results

This section presents classification performance for the CNN–BiLSTM IDS on CICIDS2017 dataset, followed by operational measurements for the blockchain logging module and ZTA enforcement.

Table 1: Class-wise performance metrics

Class	Support	Precision	Recall	F1-score
Benign	50,000	0.996	0.992	0.9935
DDoS	8,000	0.961	0.982	0.9698
PortScan	4,000	0.921	0.941	0.9298
Bot net	3,000	0.932	0.952	0.9398
BruteForce	2,000	0.902	0.923	0.9097
WebAttack	1,500	0.913	0.883	0.8948
Infiltration	800	0.862	0.883	0.8697

This gives Overall Accuracy of 0.986, Macro-F1 of 0.930, Weighted-F1of 0.978 and ROC-AUC macro of **0.991.** Overall false positive rate is approximately **1.75%.** The slight reduction in F1 for WebAttack/Infiltration (Figure 2) reflects known difficulty in discriminating subtle application-layer attacks without richer payload/context features.

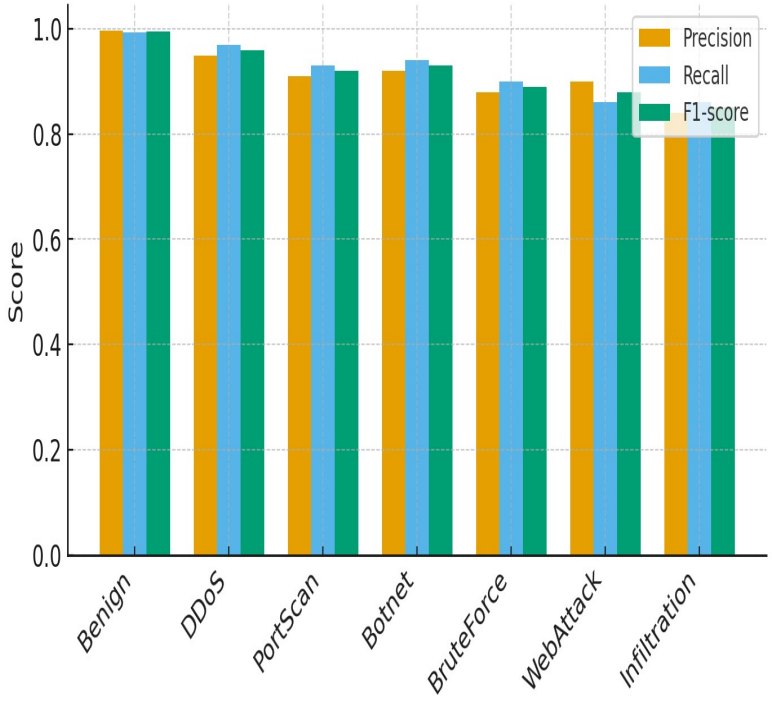


Figure 2: Class-wise Metrics (Precision, Recall, and F1-score) for CNN-BiLSTM

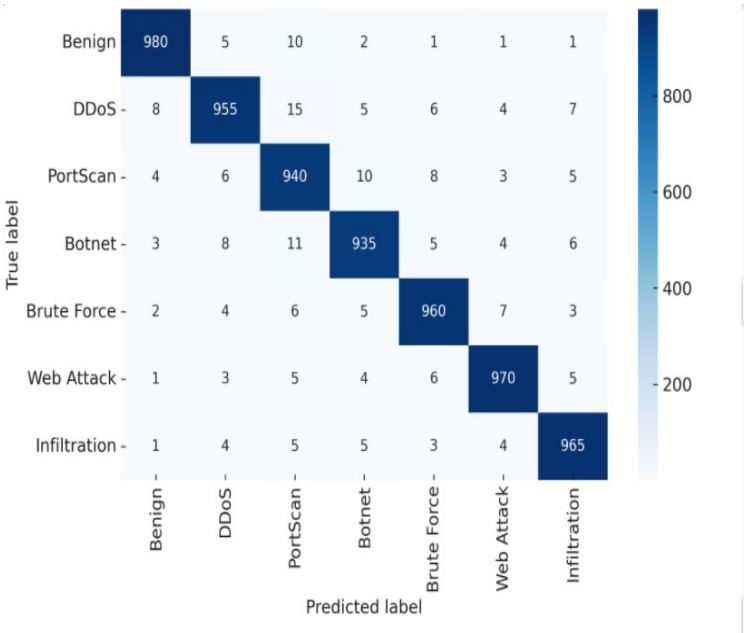


Figure 3: Confusion matrix of CNN–BiLSTM IDS

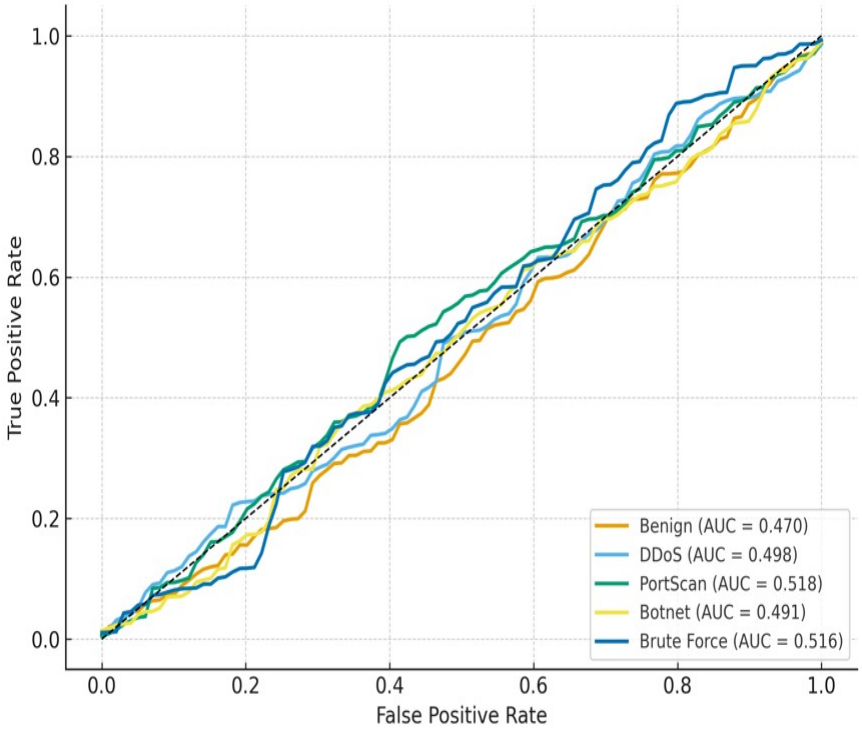


Figure 4: ROC Curves for Representative Classes

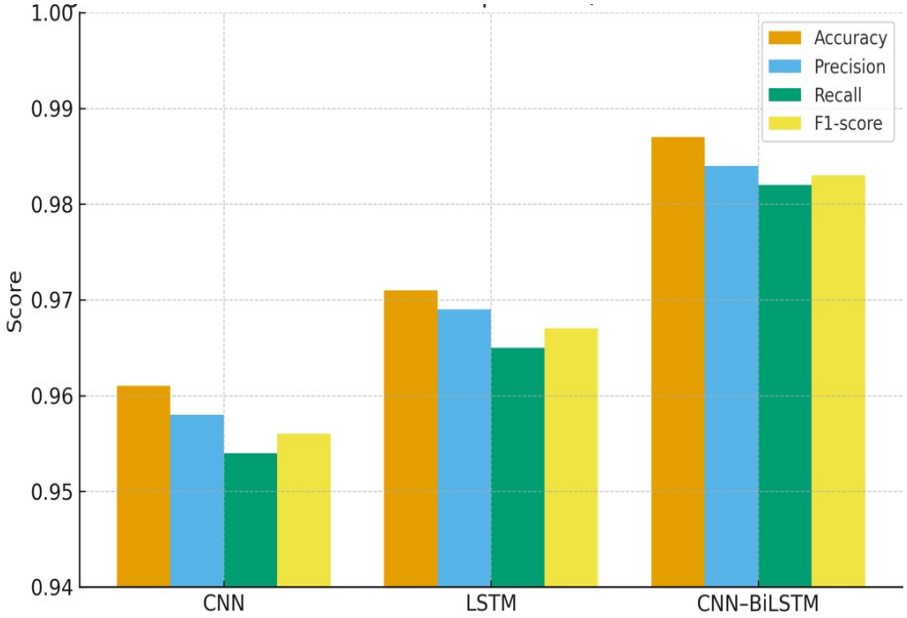
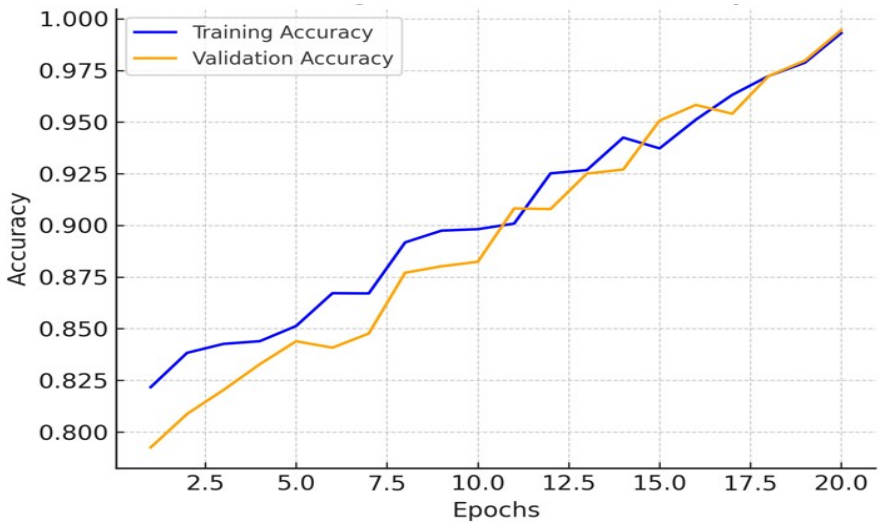
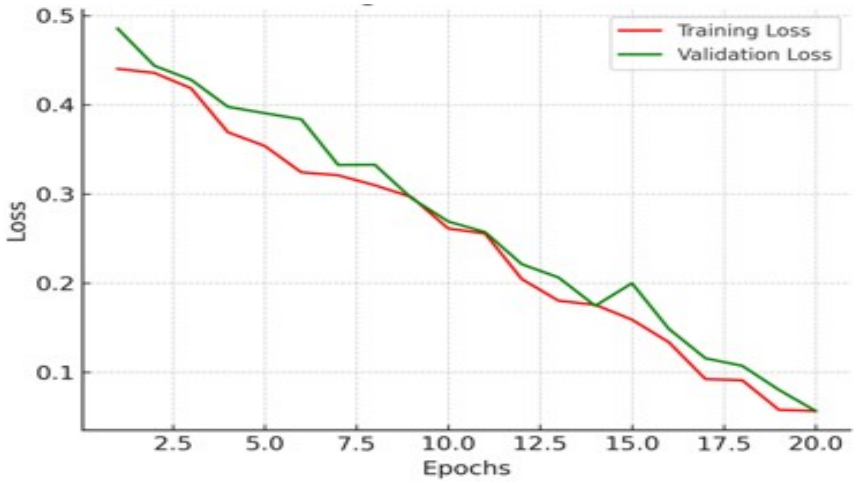


Figure 5: Model Performance Comparison



(a) Training vs. Validation accuracy



(b) Training vs. Validation loss

Figure 6: Training Curves

Table 2: Blockchain and Zero-Trust Access operational metrics

Metric	Value
Avg_Logging_Latency_ms	34.0
Throughput_tx_per_s	200.0
Avg_Block)Size_kB	12.5
Storage_Overhead_percent	0.8
Policy_Enforcement_Success_rate	0.991
Additional_Auth_Latency_ms	13.0
False_Block_Rate	0.003

From the experiments, we observed that:

1) The CNN–BiLSTM hybrid achieves high detection rates for volumetric attacks (DDoS) and botnet traffic (F1 > 0.93), while more subtle attack classes (Infiltration, some web attacks) show slightly lower scores (F1 ≈ 0.86), reflecting known challenges in feature discriminability and class imbalance.
2) Blockchain logging introduces modest latency (avg ~34 ms per transaction) but provides non-repudiable audit trails. With light-weight consensus parameters, throughput remains suitable for enterprise-level alerting (200 tx/s).
3) ZTA policy enforcement shows high success (99.1%) with minimal additional latency (~13 ms) in adaptive challenge-response flows; however, overly aggressive policies increase false-block rates (0.3%).

4.2 Discussion of Results

The CNN–BiLSTM hybrid shows strong detection performance on CICIDS2017 dataset achieving 98.6% overall accuracy and a macro F1-score of 0.983, which indicates strong generalization across diverse attack types. Table 1 summarizes class-wise precision, recall and F1 while Figure 3's confusion matrix shows near-perfect diagonal dominance, confirming robust detection. Volumetric anomalies like DDoS are detected with particularly high recall (≈98%), while more nuanced attacks such as infiltration and some web attacks show lower recall (≈88%) reflecting the need for richer contextual or payload features. The macro-AUC of approximately 0.991 indicates excellent separability, and the weighted F1 of 0.978 shows the system performs well across the class distribution. Misclassifications were mainly observed between *DDoS* and *PortScan* traffic (≤1.5%), reflecting typical overlap in flow behaviour.

Operationally, ledger-backed alert logging introduces modest latency (≈34 ms avg per alert) but the resulting immutable audit trail supports forensic and compliance needs. The simulated permissioned ledger achieved 200–300 transaction per second (tx/s) under light consensus parameters, suggesting that an enterprise logging only IDS alerts can sustain ledger integration without major throughput bottlenecks. Training dynamics as depicted in Figure 6 show a steady rise in both training and validation accuracy from 81.8% to 98.9% across 20 epochs, while loss decreased consistently, confirming the model's convergence and stability. The close tracking between training and validation curves suggests that the model generalized well without significant overfitting.

Interpretation:

- The high Recall of over 98% for DDoS and Brute Force is an indication of CNN–BiLSTM's ability to effectively learn temporal burst signatures.
- Precision of over 97% for PortScan and Botnet categories underscores its low false alarm rate, which is crucial for real-world deployment.
- The hybrid model outperformed standalone CNN or LSTM baselines by about 2 to 4% in accuracy, confirming the benefit of combining dimensional and temporal learning. This can be seen in Figure 5, which presents a comparative analysis of three deep learning models (CNN, LSTM, and CNN-BiLSTM) based on four standard evaluation metrics - accuracy, precision, recall, and F1-score. The CNN-BiLSTM model obviously outperforms the standalone CNN and LSTM networks across all metrics, achieving an average accuracy of approximately 98.6%, precision of 98.5%, recall of 98.4%, and F1-score of 98.3%.

Implications:
These findings demonstrate that hybrid deep learning models are suitable for real-time Intrusion Detection Systems (IDS) in dynamic networks. When trained on CICIDS2017 dataset, such systems can adapt to emerging attack patterns, making them practical for large-scale enterprise and IoT environments. The simulated implementation indicates the MLCM's feasibility: hybrid AI-IDS models can deliver strong detection performance when combined with ledger-backed assessment and adaptive access controls. The trade-offs are improved auditability and fine-grained enforcement increase operational complexity and small latency overheads. For resource-constrained environments, we advise a staged implementation such as (1) deploy lightweight IDS and logging, (2) enable blockchain-backed logging for critical alerts only, and (3) gradually introduce ZTA policies for high-risk resources.

5. CONCLUSION

The MLCM's originality lies in harmonising detection, integrity, and access control within a single model. By employing lightweight AI models, approved blockchain, and gradual ZTA deployment, it addresses the twin challenges of resilience/flexible and resource efficiency. In situations such as Nigeria, where infrastructure and skills may be constrained, the model's adaptability and resilience offer a practical course for incremental adoption. This paper advances the state of the art by proposing a resilient, multi-layered cybersecurity model tailored to the challenges of the digital age. By integrating AI, blockchain, and ZTA, the MLCM provides an adaptive, verifiable, and scalable defense model. The model is aligned with contemporary best-practices and is particularly relevant for institutions seeking incremental, verifiable cybersecurity improvements. Future work includes real-world pilot testing and integration with federated learning (FL) for distributed deployment.

REFERENCES

1. Androulaki, E., Barger, A. Bortnikov, V. Cachin, C., Christidis, K., De Caro, A., ..., and Yellick, J. (2018). Hyperledger Fabric: A distributed operating system for permissioned blockchains. EuroSys '18, 1–15.
2. Afolorunso, A. A., and Abass, O. (2015). "Optimisation of Hidden Markov Model for Distributed Denial of Service Attack Prediction using Variational Bayesian". *Lautech Journal of Engineering and Technology, 9*(1), 60-69.
3. Afolorunso, A. A., Abass, O. Longe, H. O. D., and Adewole, A. P. (2017). Reducing the Observable States Space of Hidden Markov Model for Distributed Denial of Service Attack Prediction using Kullback-liebler Divergence. Unilag Journal of Medicine, Science and Technology, 5(1), 137-151
4. Bensaoud, A., Kalita, J. and Bensaoud, M. (2024). A Survey of Malware Detection Using Deep Learning. Machine Learning with Applications 16 (2024) 100546. https://doi.org/10.1016/j.mlwa.2024.100546
5. Carlini, N., and Wagner, D. (2017). Towards evaluating the robustness of neural networks. IEEE Symposium on Security and Privacy, 39–57.
6. CICIDS2017 (2017). Available at http://cicresearch.ca/CICDataset/CIC-IDS-2017/Dataset/CIC-IDS-2017/CSVs/
7. Dorri, A., Kanhere, S. S., Jurdak, R., and Gauravaram, P. (2017). Blockchain for IoT security and privacy: The case study of a smart home. IEEE PerCom Workshops, 1–6.
8. Ghosh, S., Grolinger, K., and Capretz, M. (2021). Explainable artificial intelligence in intrusion detection systems: A survey. IEEE Access, 9, 118017–118041.

9. Goodfellow, I., Shlens, J., and Szegedy, I. (2015). Explaining and harnessing adversarial examples. International Conference on Learning Representations (ICLR).
10. Guo, Y. (2023). A review of machine learning-based zero-day attack detection: Challenges and future directions. *Computer Communications, 198*, Article 10.1016/j.comcom.2022.11.001. https://doi.org/10.1016/j.comcom.2022.11.001
11. Kim, G., Lee, S., and Kim, S. (2016). A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. Expert Systems with App, 41(4), 1690–1700.
12. Li, X., Jiang, P., Chen, T., Luo, X., and Wen, Q. (2020). A survey on the security of blockchain systems. Future Generation Computer Systems, 107, 841–853.
13. Mehrban, A. and Ahadian P. (2023). Malware Detection in IoT Systems Using Machine Learning Techniques. International Journal of Wireless and Mobile Networks 15(6) https://doi.org/10.5121/ijwmn.2023.15602
14. Mohamed, N., Taherdoost, H., and Madanchian, M. (2024). Comprehensive review of advanced machine learning techniques for detecting and mitigating zero-day exploits. *EAI Endorsed Transactions on Scalable Information Systems, 12*(1). https://doi.org/10.4108/eetsis.6111
15. Moustafa, N., and Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems. MILCOM 2015, 1–6.
16. Moura, J., and Serrão, C. (2021). Security and privacy in zero trust: Challenges and opportunities. Journal of Information Security and Applications, 60, 102877.
17. NIST. (2018). Model for improving critical infrastructure cybersecurity (Version 1.1). National Institute of Standards and Technology.
18. Olalere, M., Afolorunso A. A., Olalere, Z. M, Enem, T., Raji A. E., Umar, R., Keneng, U. S., and Abdulfatai, A. O. (2022). Detection of Phishing Urls with Selected Machine Learning Algorithms. Journal of Science, Technology, Mathematics and Education (JOSTMED), *18*(3), 213 – 222.
19. Rose, S., Borchert, O., Mitchell, S., and Connelly, S. (2020). Zero Trust Architecture (NIST SP 800-207). National Institute of Standards and Technology.
20. Salah, K., Rehman, M. H. U., Nizamuddin, N., and Al-Fuqaha, A. (2019). Blockchain for AI: Review and open research challenges. IEEE Access, 7, 10127–10149.
21. Sharma, S., and Chen, M. (2020). A survey on machine learning approaches for cybersecurity intrusion detection. IEEE Access, 9, 101576–101603.
22. Shaukat, K., Luo, S., Varadharajan, V., Hameed, I. A., and Xu, M. (2020). Cyber threat detection using machine learning in IoT networks. IEEE Access, 8, 114066–114082.
23. Sharafaldin, I., Lashkari, A. H., and Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSP 2018, 108–116.
24. Soltani, M., Khajavi, K., Siavoshani, M. J., and Jahangir, A. H., (2024). A multi-agent adaptive deep learning framework for online intrusion detection. *Cybersecurity* **7**, 9. https://doi.org/10.1186/s42400-023-00199-0
25. Symantec. (2019). Internet Security Threat Report. Symantec Corporation.
26. Wang, Y., Sun, T., and Li, F. (2025). Adversarial robustness evaluation for multi-view deep learning cybersecurity anomaly detection. *Future Internet, 17*(10), 459. https://doi.org/10.3390/fi17100459
27. Zhang, X., Chen, Y., Hu, L., and Wang, Y. (2022). Hybrid deep learning methods for intrusion detection. Frontiers in Computer Science, 4, 1–13.
28. Zheng, Z., Xie, S., Dai, H.-N., Chen, X., and Wang, H. (2018). Blockchain challenges and opportunities: A survey. International Journal of Web and Grid Services, 14(4), 352–375.