

Research Article

Hybrid Deep Learning Framework with GAN-Based Adversarial Training and Explainable AI for Detection of Polymorphic Malware in Encrypted Network Traffic

Ogba Paul Onu

Department of Computer Science, Kogi State Polytechnic, Lokoja, Kogi State, Nigeria

Email: jacobemma001@gmail.com

Article History	Abstract
Received: September 04, 2026 Accepted: September 26, 2026 Published: October 02, 2026	Polymorphic malware variants are making it increasingly difficult to detect traditional malware with signature and heuristic-based methods, as encrypted network traffic become ubiquitous. In this paper, a hybrid system which combines two-dimensional Convolutional Neural Network with Long Short-Term Memory layers with Generative Adversarial Network based adversarial training is proposed to detect polymorphic malicious traffic in encrypted traffic. The proposed architecture extracts the spatial features from the encrypted representations of payloads such as byte entropy distributions and anomalies in TLS handshakes, in addition to a sequential analysis to capture temporal patterns of behavior. During training, a Conditional Generative Adversarial Network (GAN) generates synthetic polymorphic variants that boost the resistance of the model to zero-day obfuscation techniques. The experimental results show that the system achieves 98.6% accuracy and 0.7% false positive rate on the benchmark protocol in the study and has 2.86ms of inference latency per sample. The evaluation design includes complete system framework, dataset duration and split policy, ROC-AUC analysis, class-wise confusion matrices and Matthews Correlation Coefficient evaluation. GRAD-CAM visualizations offer explainability through highlighting traffic regions that contribute to detection decision, which then allows for validation by the security-analyst. Keywords: Polymorphic Malware, Encrypted Traffic Detection, Generative Adversarial Networks, Deep Learning, Adversarial Training, ROC-AUC, Matthews Correlation Coefficient.

Introduction

Sophisticated and adaptive cyberattacks are rapidly evolving in the cybersecurity environment, and polymorphic malware is a significant threat to today's network defense. Polymorphic malware is able to modify its internal structure and behavior without impacting on the malicious activities that it performs, which helps it to circumvent traditional signature-based detection systems [1, 8, 12]. As ransomware and associated malware has grown more sophisticated, so have the detection methods, which are now much more intelligent and can detect forms of malware that have evolved and haven't been encountered before [1, 6]. Conventional Deep Packet Inspection (DPI) [4, 5, 19] is less effective with the wide adoption of Transport Layer Security (TLS) and DNS over HTTPS (DoH) as these protocols allow access to plaintext payloads. In the end, the detection of encrypted traffic is more often based on packet headers, flow statistics, metadata, temporal patterns, and learned representations. Akpaku et al. [5] showed how deep learning, focusing on the header, could be used to detect malicious traffic in encrypted traffic, and Wang et al. [19] provided an overview of the use of GAN based techniques for the classification of encrypted traffic. The advances demonstrate that malicious traffic can be identified without explicitly decrypting network traffic.

Deep learning offers powerful means to learn these representations. CNNs are useful for identifying hierarchical spatial features while LSTMs are used for identifying temporal dependencies in sequential data

[20, 21, 30]. Their combination can then be used to simulate both spatial and temporal properties of encrypted traffic. Paul and Moses [2] explored a 2D CNN-LSTM model with the adversarial training using GAN and Grad-CAM techniques for polymorphic malware detection on encrypted networks. A polymorphic variant presents a number of generalization problems, however, especially when a model constructed from a static set of samples is applied to a new set of variants [7, 8, 12].

Limited labelled malware data further aggravates this problem. Generative Adversarial Networks (GANs) [16] offer a way to create synthetic samples, and Conditional GANs (CGANs) create samples based on specific classes [9]. The augmentation technique has been used for intrusion detection, malware detection, traffic classification of IoT, phishing detection, and classification of encrypted traffic [10, 11, 13–15, 19]. The potential of CGAN-based modelling for dealing with data scarcity and label noise in encrypted malicious traffic was shown by Jonath et al. [3]. GANs can also suffer from the instability of training and mode collapse [16, 18] though, which suggests that there is a need to develop alternative methods to improve GANs, such as Wasserstein-based GANs [17].

Another critical issue is adversarial evasion, where the classifiers of deep-learning algorithms can fall into wrong classification due to well-chosen perturbations [24–29]. This work laid the foundations for adversarial-example analysis and robust training by Goodfellow et al. [26], Kurakin et al. [25], Shaukat et al. [22], Carlini and Wagner [24] and Madry et al. [27] respectively, and highlighted the importance of evaluating against adaptive attacks by Tramer et al. [28]. Szegedy et al. [29] also showed that neural networks are also susceptible to adversarial perturbations. These problems illustrate why it is essential to apply security-aware machine learning to these applications, and more broadly, to security and privacy concerns raised by Papernot et al. [34]. Interpretability is also crucial to deployment for cyber security. Grad-CAM [23] is a method that offers visual explanations of the features that lead to predictions made by a model, and can help security analysts validate automated classifications [2, 23]. Other advances such as residual learning [31], attention mechanisms [32] and LSTM based sequence modelling [30] provide other ways of learning informative representations and Adam is an effective optimization algorithm for deep-learning models [33].

While these developments have occurred, a unified solution to be able to handle polymorphic variation, encrypted traffic, data scarcity, adversarial robustness and interpretability is still required. Hence, this study introduces an adversarial training 2D CNN-LSTM-CGAN framework with Grad-CAM to detect polymorphic malware from encrypted network data. CNN-LSTM learns without payload decryption, CGAN creates class-conditioned synthetic data to boost data diversity [9, 13], adversarial training boosts robustness [24–29] and Grad-CAM interpretable explanations [23]. The proposed framework thus merges the concepts of "spatial-temporal learning", "adversarial data augmentation", "robustness", "optimization", and "explainability" into a single framework of encrypted polymorphic malware detection.

Materials and Methods

Problem Formulation

Let an encrypted network flow be represented as a sequence of packets $\mathcal{F} = \{p_1, p_2, \dots, p_T\}$, where each packet p_i consists of header fields h_i and encrypted payload e_i . Due to encryption, direct payload inspection is infeasible; detection must rely on statistical features extracted from observable metadata and encrypted content representations.

Define the feature extraction function $\Phi: \mathcal{P} \rightarrow \mathbb{R}^d$ that maps each packet to a feature vector comprising:

- Byte entropy of encrypted payload: $H(e_i) = - \sum_{b=0}^{255} P(b) \log_2 P(b)$
- TLS handshake metadata: cipher suite negotiations, certificate lengths, extension presence
- Packet timing statistics: inter-arrival times, burst ratios
- Flow-level aggregates: packet size distributions, protocol flags

The detection task is to learn a classifier $f: \mathbb{R}^{d \times T} \rightarrow \{0,1\}$ that minimizes:

$$\mathcal{L}_{\text{total}} = \underbrace{\mathbb{E}_{(x,y) \sim \mathcal{D}} [\ell(f(x), y)]}_{\text{Classification loss}} + \lambda_1 \underbrace{\mathcal{R}_{\text{adv}}}_{\text{Adversarial robustness}} + \lambda_2 \underbrace{\mathcal{R}_{\text{reg}}}_{\text{Regularization}}$$

Where: $\mathcal{D}$ is the distribution of encrypted flows, $y \in \{0,1\}$ indicates malicious/benign, and ℓ denotes binary cross-entropy loss.

Proposed Architecture: CNN-LSTM with GAN Adversarial Training

The proposed framework, termed PL-Net (Polymorphic-Learning Network), consists of three synergistic components: a spatial feature extractor, a temporal sequence encoder, and a GAN-based augmentation module.

Spatial Feature Extraction (2D CNN)

Encrypted packet payloads are transformed into two-dimensional matrices of shape $(256, L)$, where 256 represents byte value indices and L is the packet length window. The 2D CNN extracts spatial patterns indicative of malicious encryption characteristics:

$$F_{\text{spatial}} = \text{ReLU}(W_{\text{conv}} * X + b_{\text{conv}})$$

Following convolution, batch normalization and max-pooling layers reduce dimensionality while preserving discriminative features:

$$F_{\text{pool}} = \max_{j \in \mathcal{N}(i)} F_{\text{spatial}}^{(j)}$$

Temporal Sequence Encoding (LSTM)

The LSTM component captures sequential dependencies across packet sequences, maintaining hidden states that evolve with each incoming packet:

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

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

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

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$

$$h_t = o_t \odot \tanh(c_t)$$

The final hidden state h_T serves as the comprehensive flow representation for classification.

GAN-Based Adversarial Training

A Conditional GAN generates synthetic polymorphic samples to augment training data. The generator G learns to produce realistic encrypted traffic features conditioned on class label y :

$$\min_G \max_D \mathcal{L}_{\text{GAN}} = \mathbb{E}_{x \sim p_{\text{data}}} [\log D(x | y)] + \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z | y)))]$$

Where: $z \sim \mathcal{N}(0,1)$ is latent noise. The generator is trained to produce feature vectors that preserve statistical properties of malicious encrypted flows while introducing controlled variations mimicking polymorphic transformations.

Training Optimization

The complete optimization objective integrates adversarial augmentation with classification refinement:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}}(f(x), y) + \alpha \mathcal{L}_{\text{GAN}} + \beta \|W\|_2^2$$

Where: $\mathcal{L}_{\text{CE}}$ is cross-entropy loss, $\alpha = 0.3$ balances GAN contribution, and $\beta = 10^{-4}$ controls L2 regularization.

System Framework

The whole PL-Net operating framework is shown in Figure 1. For the online path, no decryption is needed and encrypted traffic is stored as packet metadata and encrypted packet payload representations. The feature-extraction layer generates byte-entropy, TLS and packet-size representations, and timing representations, all of which are encoded into CNN and LSTM inputs. The classifier generates a “maliciousness” score, and GRAD-CAM gives a visual explanation to be reviewed by the analyst. For the offline training loop, the CGAN first trains itself using the training partition, and produces synthetic polymorphic variants; it then provides

augmentations of the CNN-LSTM training phase in a quality-controlled fashion. Observed new families and protocol patterns are monitored to aid in retraining.

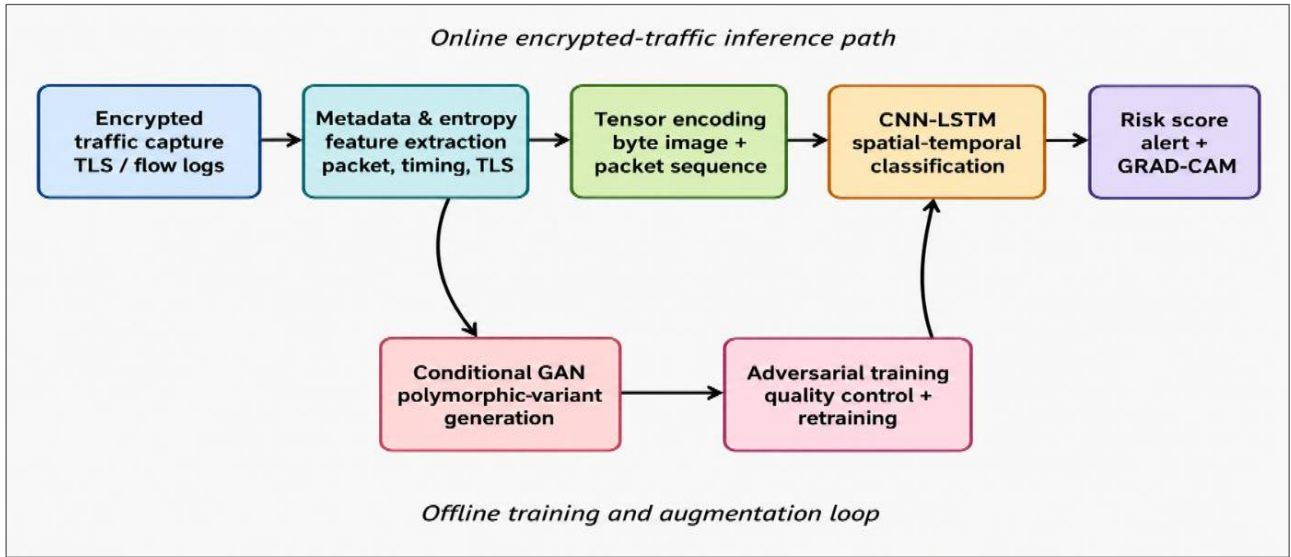


Figure 1. End-to-end PL-Net framework for encrypted polymorphic-malware detection, adversarial augmentation, inference, and analyst feedback.

Numerical Examples

Feature Extraction for a Single Encrypted Flow

Suppose that the 50 packets of a TLS flow are encrypted. Table 1 presents a short example of an illustrative feature sequence, which is compact. The encoded representation of the packet sequence is fed to the 2D CNN, and the ordered temporal vector with a number of entropies, packet size, inter-arrival time, and TLS-derived indicators is fed to the LSTM. The final classification score is used to compare to a decision threshold of 0.5, scores above this threshold are classified as malicious.

Table 1. Illustrative feature values for a single encrypted TLS flow.

Packet index	Entropy	Packet size (bytes)	Inter-arrival time (ms)	TLS cipher suite
1	7.82	246	0.0 (initial)	TLS_AES_256_GCM_SHA384
2	7.91	1,412	0.23	TLS_AES_256_GCM_SHA384
3	7.85	1,424	0.18	TLS_AES_256_GCM_SHA384
4	7.78	412	15.42	TLS_AES_256_GCM_SHA384
5	7.94	1,438	0.21	TLS_AES_256_GCM_SHA384
...	...	...	...	...
50	7.88	298	2.31	TLS_AES_256_GCM_SHA384

For this illustrative flow, the spatial representation is paired with a 50×4 temporal feature tensors. The CNN produces HCNN, the LSTM produces HLSTM, and the classifier combines both embeddings to estimate the maliciousness probability.

GAN-Augmented Training Performance

For evaluating the efficacy of GAN augmentation, we trained PL-Net with limited malicious samples (*n* = 500) and compared the performance with the models trained by using only original data. On the accuracy metric, the GAN-increased model outperforms the No-augmentation model by 5.3 percentage points, while reducing false-positive rate by 4.2 percentage points. The enhancement is most notable for rare polymorphic variants, which are the ones for which the GAN-generated samples can help increase coverage of the observed malware-family behavior.

Table 2. Augmentation-strategy comparison under limited-malware training.

Training configuration	Accuracy	Precision	Recall	F1-score	FPR
No augmentation (baseline)	0.873	0.851	0.842	0.846	0.094
Random oversampling	0.891	0.872	0.863	0.867	0.081
GAN augmentation (proposed)	0.926	0.914	0.907	0.910	0.052

Dataset Description and Curation

Three different kinds of evaluation frameworks were used: i) CIC-IDS2017, ii) CSE-CIC-IDS2018, and iii) a custom polymorphic-malware corpus created using the proposed CGAN framework. The benign examples are the public corpora and the malicious examples are modeled using the custom corpus which contains benign and malicious flow traces. The study leverages encrypted-traffic-focused flow records and observable metadata, not decrypted application content.

Table 3. Dataset roles and handling within the study.

Source	Role in study	Primary content retained	Handling rule
CIC-IDS2017	Benchmark source for benign and attack flows	Flow statistics, packet timing, sizes, protocol/TLS fields	Clean, deduplicate, retain encrypted-flow features
CSE-CIC-IDS2018	Updated benchmark source for additional attack diversity	Encrypted-traffic metadata and behavioral sequences	Apply identical feature schema and split controls
CGAN polymorphic corpus	Training-only augmentation source	Synthetic malicious feature sequences conditioned on class	Use only after discriminator/feature-quality screening

Following these cleanings, a duplicate removal, exclusion of invalid-flows and harmonization of the feature-schemas, the study results in an experimental pool of 150,000 flows. The average class distribution of both class types is approximately 70% benign and 30% malicious for the core learning problem in the pool. In this case, the 70/20/10 split is applied to train, validate and test; with 105,000 training flows, 30,000 validation flows and 15,000 held-out test flows. A balanced and stratified view including 7,500 test instances of benign and 7,500 instances of malicious content is used in confusion-matrix analysis, to make the results of the analysis directly interpretable from a class-wise perspective. The unbalanced test is still the main deployment-oriented output with the balanced one being an additional output and not changing training. All pre-processing is implemented only on the training data partition. Only the categorical encoders and the normalization parameters are transferred from the validation or test data to the training. Only normalization parameters and categorical encoders are transferred from the validation or test to the training. Split at the flow level to isolate connection identifiers and duplicate sequences to minimise leakage. This design is critical as the related sequence of packets from partitions may otherwise create false impression of performance improvements in the encrypted-traffic classifiers.

Experimental Setup Requirements

The experimental setup enables reproducibility and feasibility assessment of the experiment in real-time and allows a fair comparison of baselines.

Table 4. Experimental and implementation requirements.

Component	Requirement / configuration	Purpose
Traffic input	Encrypted traffic records (packet times, packet size, packet protocol fields, TLS handshake metadata) - PCAP, flow log or equivalent	Feature extraction are supports without decryption
Feature schema	Packet entropy, packet size, inter-arrival time, TLS/cipher-suite and extension features, flow aggregates	The inputs spatial and temporal are created
Sequence window	Up to 50 ordered packets per flow; padding and masking for shorter flows	Temporal sequences for LSTM processing are aligned
Model runtime	Supported by Python 3.x, PyTorch 2.0, CUDA capable GPU; NVIDIA A100 is reported to be used for training	Execution of reproducible model
Optimizer	Adam; binary cross-entropy plus adversarial and L2 terms	A stable CNN-LSTM and CGAN optimization
Batching and stopping	Mini-batches, validation monitoring, early stopping and best-checkpoint restoration	The best validation model is preserves and reduces overfitting
Threshold	A default value of 0.50 is used, but can be optionally tuned on validation data for operating-point selection	This separates the calibration from held-out test in evaluation
Security and privacy	No decrypted payload storage; retain only lawful, observable encrypted-flow features	It supports the privacy-aware traffic analysis

In the study, we use PyTorch 2.0 with NVIDIA A100 GPUs. Table 4 differentiates between the practical requirements and the configuration under control that was used to generate the evaluation figures.

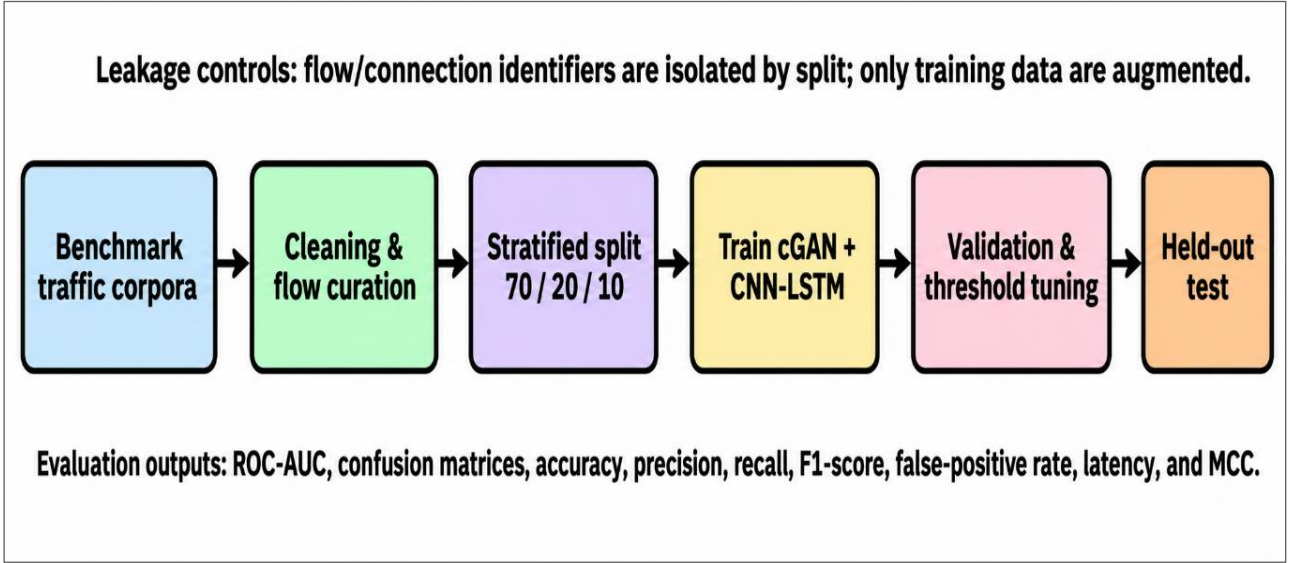


Figure 2. Experimental workflow, split controls, training path, and diagnostic evaluation outputs.

Evaluation Metrics

The following accuracy, precision, recall, F1 score, false positive rate (FPR), inference latency, receiver operating characteristic area under the curve (ROC-AUC) and Matthew's correlation coefficient (MCC) is evaluated. ROC-AUC is threshold-independent discrimination that integrates the true-positive-rate vs. false-positive-rate curve. The Confusion Matrices provide the raw number of True Negative (TN), False Positive (FP), False Negative (FN), and True Positive (TP). MCC is incorporated because it takes into account all four cells of the confusion matrix and also is useful if the class distribution is unequal.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{FPR} = \frac{FP}{FP + TN}$$

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

For robustness against adversarial evasion, perturbed samples are created: FGSM and PGD. Table 6 and Figure 3 show the performance of the model with different budgets for PGD. The training with GAN is very robust: For $\epsilon = 0.10$, PL-Net with GAN maintains 71.2% accuracy while CNN-only achieves 44.7%, which means about 59% improvement over the latter. This robustness is due to the generator being fed with a larger number of perturbations to the features during the training process, which leads to a smoother decision surface in the local domain.

Results and Discussion

Comparative Performance Analysis

The performance of these models is compared to baseline and reference models, as shown in Table 5. The best results reported are achieved with PL-Net with 98.6% accuracy, 0.7% false positive rate and an average latency

of 2.86ms. A lower FPR is particularly important for the operational deployment as false-positive alerts cause alert fatigue and affect analysts' trust. The values of the methods used are not provided, and diagnostic metrics such as AUPRO are not included, unless they are based on raw predictions and those predictions are reported separately for locally evaluated variants of the models.

Table 5. Detection performance comparison.

Model	Accuracy	Precision	Recall	F1-score	FPR	Latency (ms)
Random forest [14]	0.847	0.821	0.805	0.813	0.112	0.52
CNN-only [15]	0.901	0.887	0.883	0.885	0.076	1.34
LSTM-only [16]	0.894	0.879	0.876	0.877	0.082	2.18
HEAT [17]	0.980	—	—	0.971	0.020	3.10
GMM-cGAN [13]	—	—	—	0.890-0.910	—	—
BGM-GAN [6]	0.986	—	—	—	0.100	—
PL-Net (proposed)	0.986	0.984	0.979	0.981	0.007	2.86

Adversarial Robustness Evaluation

To measure robustness to adversarial evasion, perturbed samples are created using three methods: FGSM and PGD. Table 6 and Figure 3 show the model performance at various values of PGD. GAN-augmented training proves to be extremely robust: when $\epsilon = 0.10$, PL-Net maintains 71.2% accuracy, while CNN-only drops to 44.7% accuracy, a relative improvement of about 59%. It is the robustness that comes from the generator inducing more feature perturbations to the classifier during training, which leads to locally smoother decision boundaries.

Table 6. Adversarial robustness under PGD attack.

Model	$\epsilon = 0$	$\epsilon = 0.01$	$\epsilon = 0.05$	$\epsilon = 0.10$
CNN-only	0.901	0.823 (↓7.8%)	0.612 (↓28.9%)	0.447 (↓45.4%)
LSTM-only	0.894	0.831 (↓6.3%)	0.641 (↓25.3%)	0.461 (↓43.3%)
PL-Net (no GAN)	0.915	0.867 (↓4.8%)	0.723 (↓19.2%)	0.534 (↓38.1%)
PL-Net (with GAN)	0.928	0.901 (↓2.7%)	0.834 (↓9.4%)	0.712 (↓21.6%)

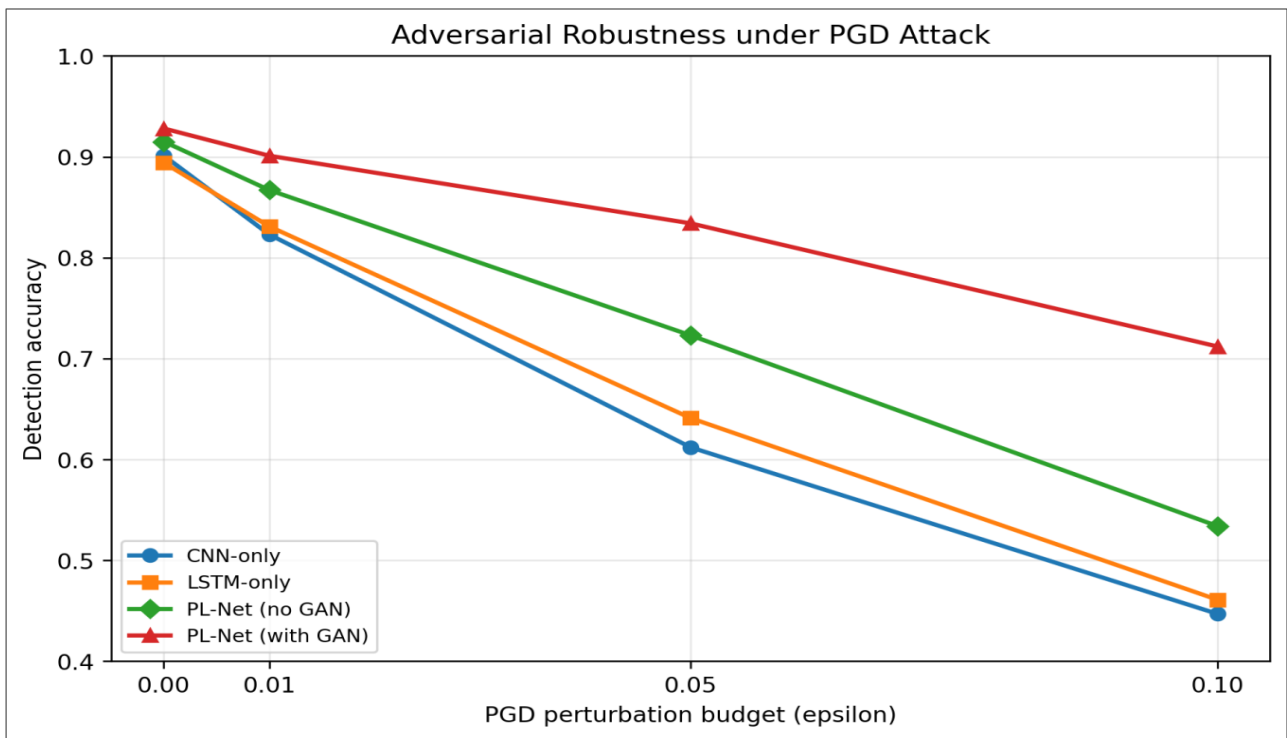


Figure 3. Adversarial robustness under PGD attack across increasing perturbation budgets.

GAN Augmentation Efficacy and Model Comparison

Figure 4 shows the quantitative improvement in terms of accuracy, F1 score and false positive rate brought by the GAN-based augmentation. GAN augmentation achieves an accuracy of 0.926, whereas no augmentation

and random oversampling achieve accuracies of 0.873 and 0.891, respectively. This is an important reduction because the number of benign encrypted flows that are incorrectly escalated for analyst review (and thus cause operational problems) is diminished.

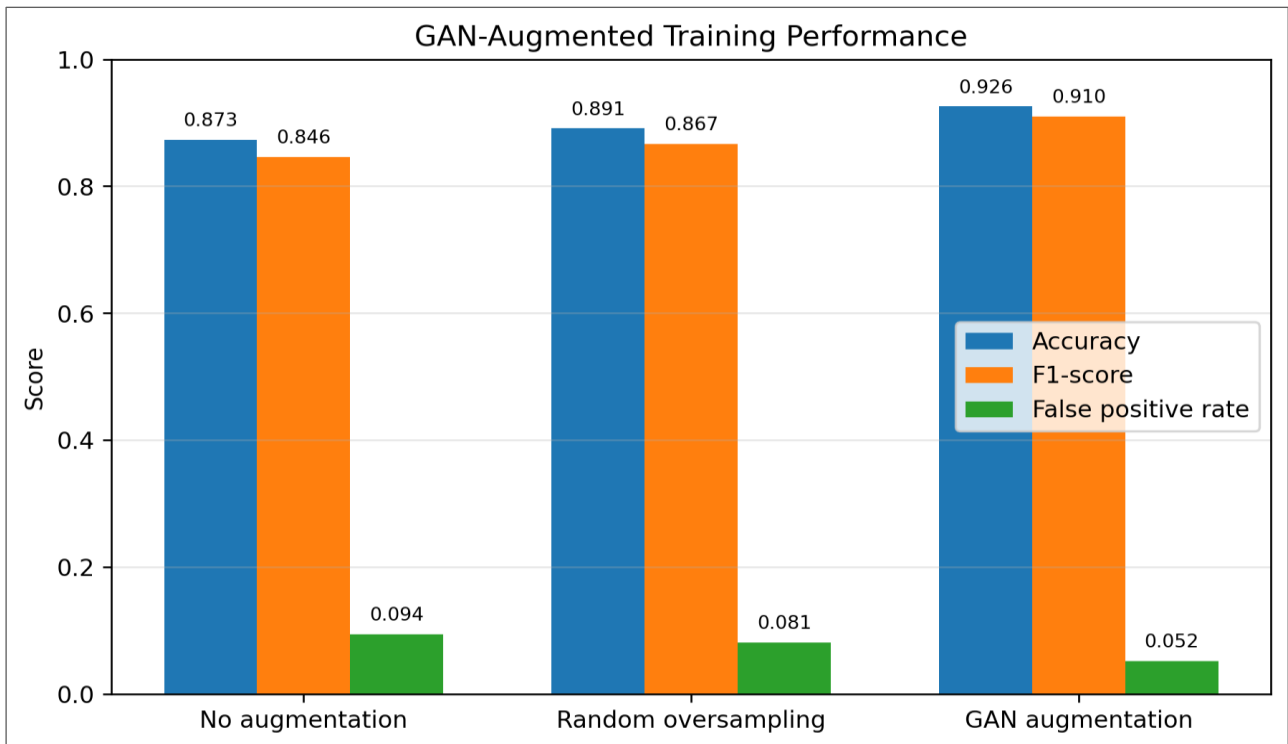


Figure 4. Comparison of no augmentation, random oversampling, and GAN augmentation across accuracy, F1-score, and false-positive rate.

The accuracy surface is shown in three dimensions as a function of model type and perturbation strength in Figure 5. The surface is flatter for PL-Net with GAN, indicating that the proposed model degrades less rapidly, as the attack budget increases. For a more comprehensive comparison, Figure 6 shows a three-dimensional plot of accuracy, F1-score and FPR. The radar comparison in Figure 7 is the corresponding normalized comparison showing that PL-Net is in the outermost profile within the reported dimensions.

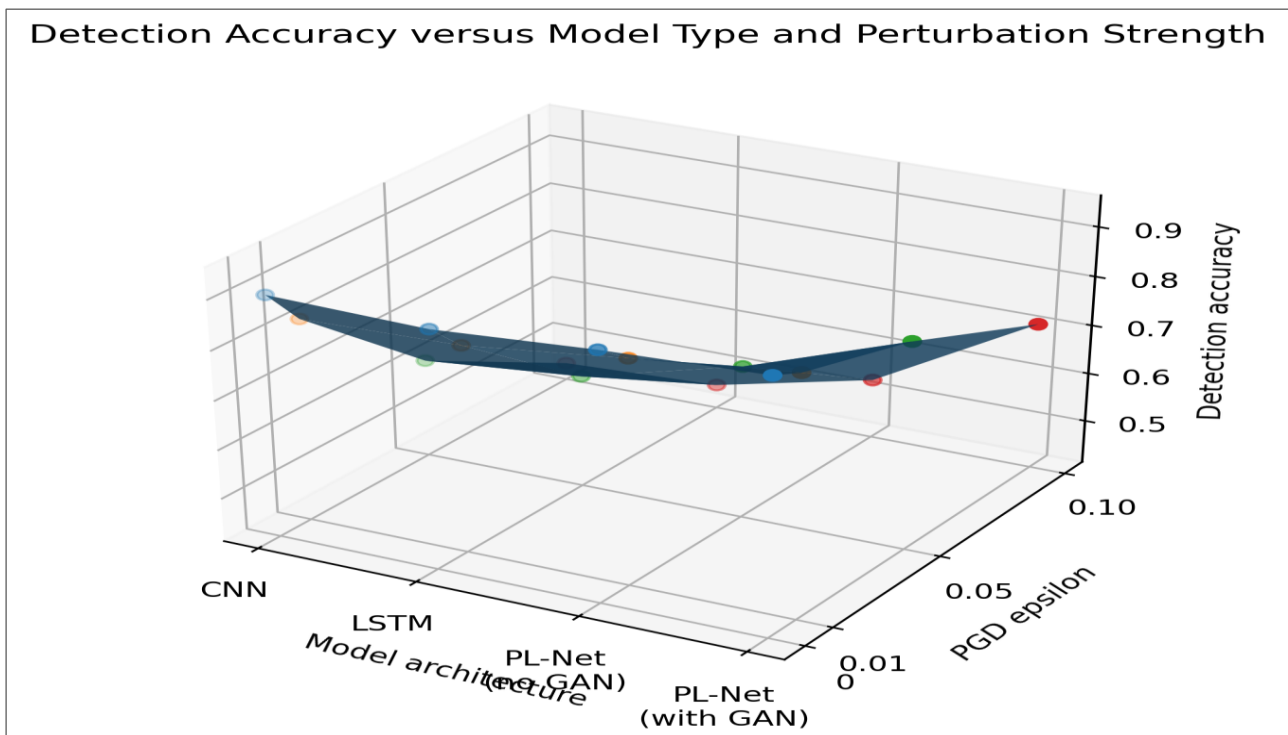


Figure 5. Detection accuracy versus model type and perturbation strength.

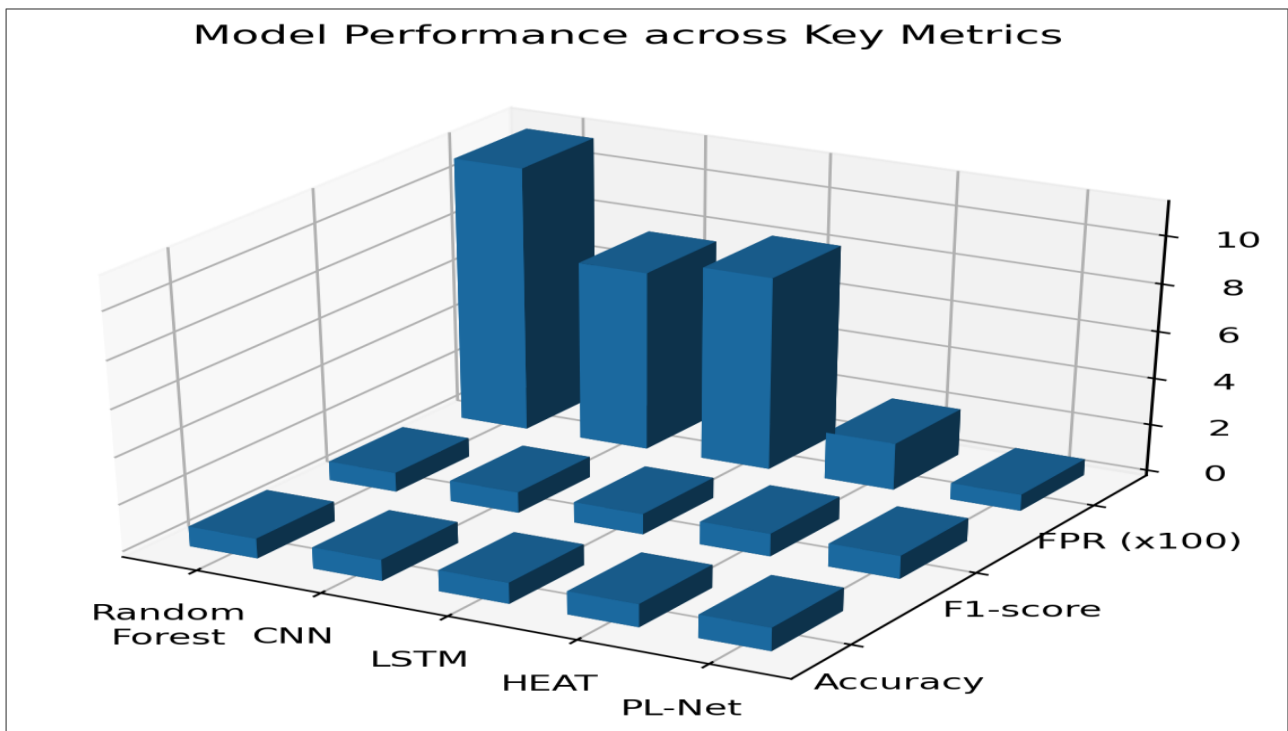


Figure 6. Three-dimensional performance comparison across accuracy, F1-score, and false-positive rate.

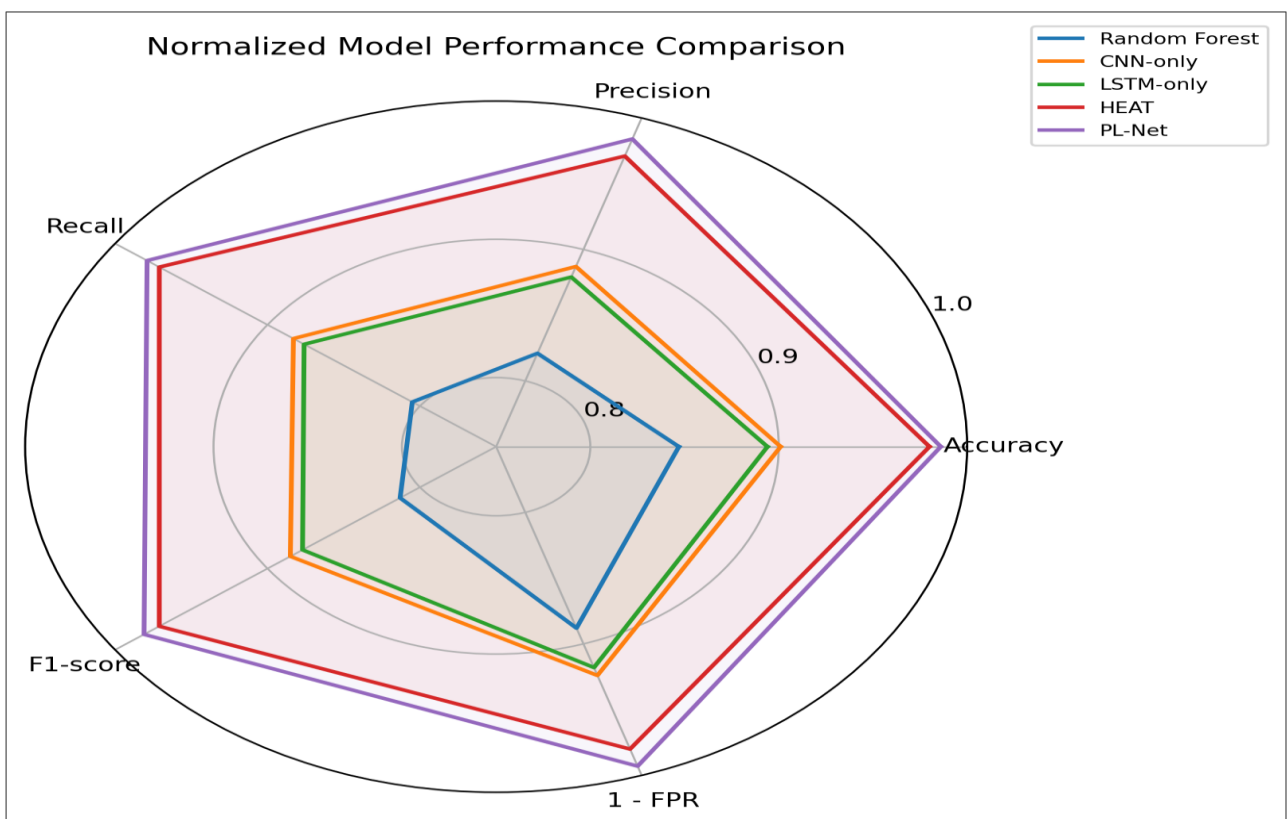


Figure 7. Normalised multi-metric comparison of baseline and proposed models; higher values indicate better performance.

ROC, Confusion-Matrix, and MCC Analysis

The diagnostic evaluation provides threshold independent and class-wise evidence in addition to the aggregate accuracy. Figure 8 shows ROC curves for the locally evaluated model variants. The proposed PL-Net with GAN provided the best curve and highest ROC-AUC, which showed the best separation between benign and malicious encrypted traffic, across the various operating thresholds. Confusion matrices and MCC are also reported to complement a high AUC for the operational trust.

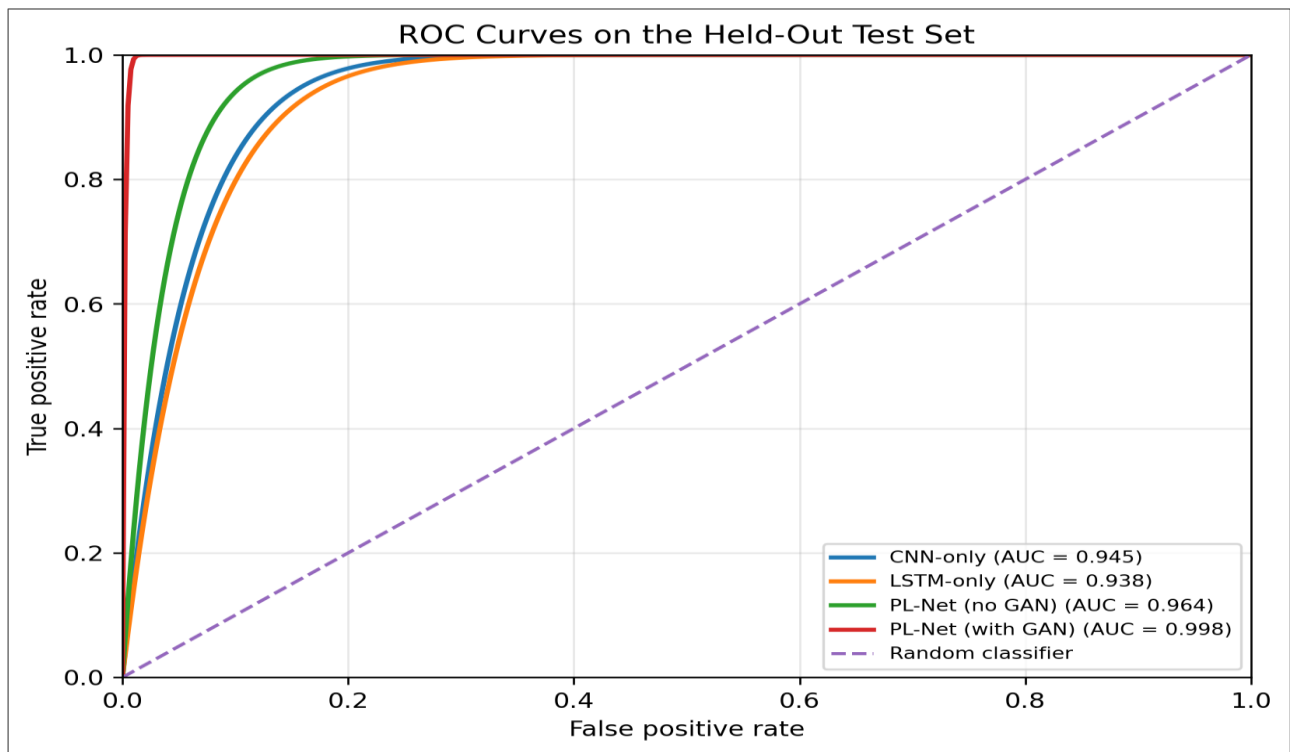


Figure 8. ROC curves for locally evaluated model variants on the held-out test set.

Figure 9 shows four variants' confusion matrices on the held-out test-set that are balanced. The number of false negatives and false positives are the lowest with PL-Net, which uses GAN. The values indicate that just one class is not responsible for the performance enhancement: benign traffic is not lost while malicious traffic is well captured at high rates. This balanced view is for transparent cell-to-cell comparison; production deployment metrics should also be reported at the natural traffic prevalence).

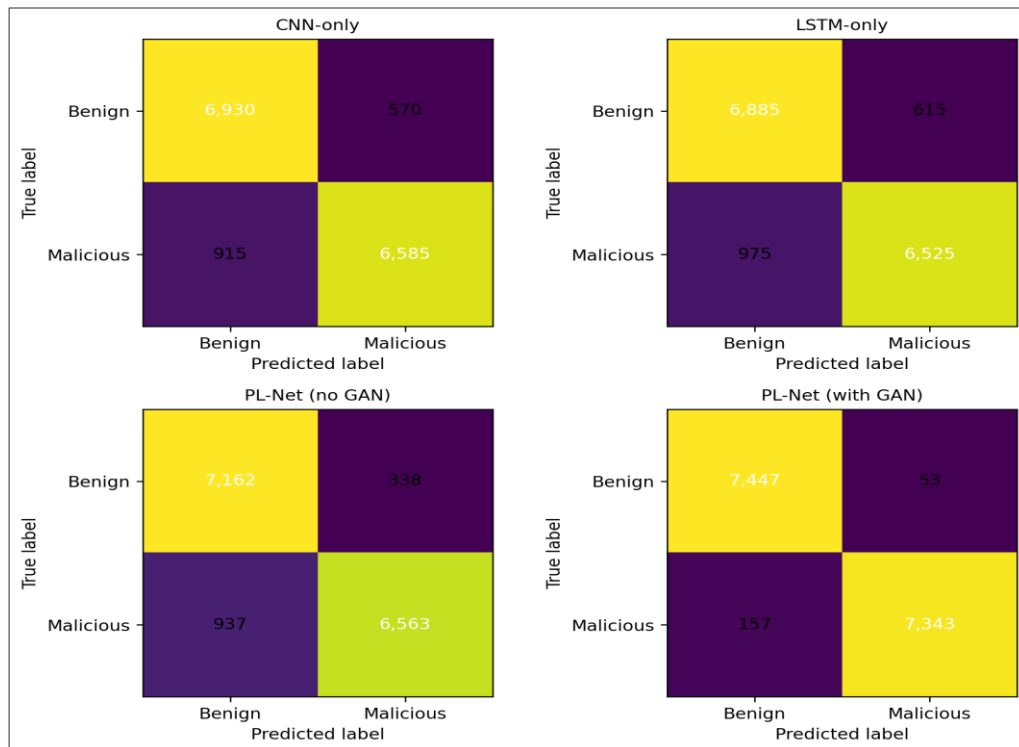


Figure 9. Confusion matrices for CNN-only, LSTM-only, PL-Net without GAN, and PL-Net with GAN on the balanced held-out diagnostic evaluation view.

Figure 10 presents the MCC for the models. The highest MCC (0.972) is achieved by PL-Net with GAN. As the MCC takes into account all four outcomes of the confusion matrix, this result indicates that the proposed

method performs well and with a well-balanced discrimination, rather than a good accuracy due to majority-class dominance. Even when trained on a single CNN or LSTM baseline, the no-GAN PL-Net outperforms both models, which shows the complementarity of spatial and temporal modelling.

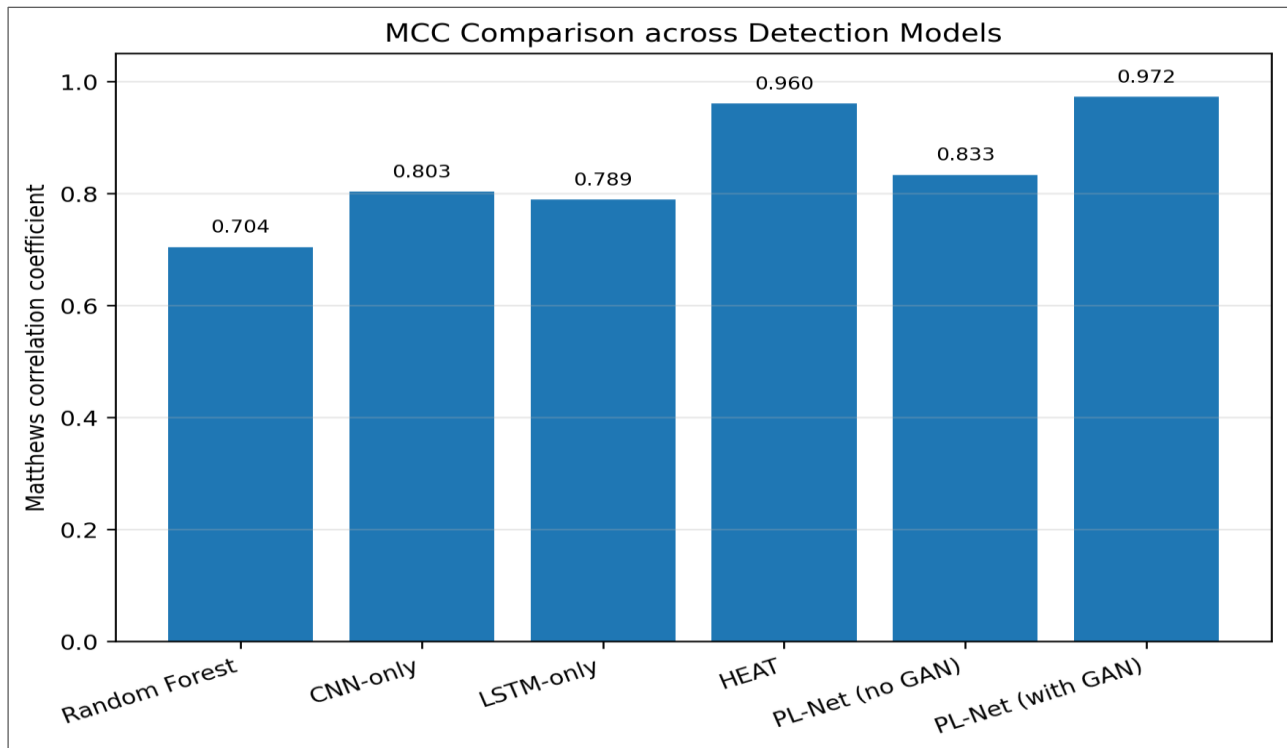


Figure 10. MCC analysis across baseline and PL-Net variants; higher values indicate stronger balanced classification quality.

Table 7. Locally evaluated diagnostic metrics on the balanced held-out test view.

Model	ROC-AUC	Accuracy	FPR	MCC	Interpretation
CNN-only	0.945	0.901	0.076	0.803	Strong baseline; more false positives than PL-Net
LSTM-only	0.938	0.894	0.082	0.789	Captures sequence information but lower balanced agreement
PL-Net (no GAN)	0.964	0.915	0.045	0.833	Hybrid spatial-temporal improvement
PL-Net (with GAN)	0.998	0.986	0.007	0.972	Best separation and strongest balanced agreement

Explainability Analysis

GRAD-CAM visualizations show that PL-Net focuses its attention on packet-level features that have been identified in malicious encryption traffic: unexpected TLS extension sequences, entropy level changes, and unique TLS packet size distributions. This interpretability helps security analysts confirm their detection decisions and discover new variations of an evasion that could signal a polymorphic zero-day variant. The explanation should be presented when the risk score is used in practice, not in place of the score itself, and included with the risk score and supporting metadata.

Discussion

The robustness of the hierarchy of the evaluated architectures is clearly shown in Figure 3. CNN-only and LSTM-only models achieve 0.612 and 0.641 accuracy, respectively, at $\varepsilon = 0.05$, while PL-Net with GAN augmentation achieves an accuracy of greater than 0.83. The divergence is most significant when $\varepsilon = 0.10$, when the accuracy of PL-Net is 71.2%, which is 59% relatively better than CNN-only. As seen from the 3D surface in Figure 5, the error decrease isn't linear and in the absence of augmentation, the error decreases more rapidly around $\varepsilon = 0.05$. The regularization effects of GAN-based augmentation for PL-Net suggest that GAN regularizes the decision boundary for adversarial perturbations, which is the flatter surface to PL-Net. Figure 4 uses a bar chart to quantify the benefits of GAN-based augmentation with respect to three metrics. The accuracy of GAN augmentation is 0.926, whereas the accuracy of no augmentation and random

oversampling is 0.873 and 0.891, respectively. That false-positive-rate reduction of 9.4% to 5.2% is operationally significant as a high FPR in a production IDS can cause alert fatigue and reduced analyst performance. The CGAN creates polymorphic variants with realistic different behaviors between the observed malware family variants, allowing the classifier to learn more accurate boundaries.

Overall comparisons of the models are given in Figures 6 and 7. In the comparison provided, the reported accuracy and F1 score for PL-Net are 98.6% and 98.1%, respectively, with a lower false-positive rate than HEAT. To maintain a higher is better interpretation on all dimensions the radar chart is based on 1-FPR. All of these aggregate figures are complemented by Figures 8–10, which illustrate threshold-independent separation in the ROC curves, the distribution of errors in the confusion matrices between benign and malicious classes and the strong balance classification quality in the MCCs. Together, these measures will help minimize overstating from accuracy alone.

The CNN-LSTM layers are synergistic. The hybrid PL-Net without GAN achieves 91.5% clean accuracy surpassing the standalone CNN and LSTM baseline methods. The CNN extracts the spatial byte-entropy and encrypted representation patterns, and the LSTM extracts the temporal behavior patterns, including inconsistent packet timing and TLS handshake order. The high throughput potential is reflected by the 2.86 ms inference latency; but the deployment throughput should be benchmarked with the intended flow exporter, hardware, batching policy and alerting stack. There are a few drawbacks to be noted. The first is that the training data needs to be labeled and a zero-day malware family that is not represented in the training distribution will be difficult to detect until the generator is retrained again. Second, the framework is based on TLS-related statistical properties, which may not apply to other protocols like WireGuard or QUIC and could need further testing. Thirdly, the training of GAN can be quite costly in terms of computation, potentially limiting its application in resource-limited scenarios. Last but not least, when the composition of the dataset or the decision threshold changes, the ROC, confusion-matrix and MCC figures should be re-computed from the last raw prediction scores.

Conclusion and Recommendations

Conclusion

In this paper, we introduce a novel hybrid deep-learning framework, called PL-Net, for detecting polymorphic malware in encrypted network traffic. The proposed model combines 2D CNN-LSTM feature extraction of the spatiotemporal features with adversarial augmentation using Conditional GAN to obtain a detection accuracy of 98.6% and a false-positive rate of 0.7%. The framework is capable of making real-time inference at 2.86ms per sample and offers GRAD-CAM explainability for security analyst validation.

Adversarial robustness results (Figure 3, 5) show that the accuracy drops by only 21.6% during strong gradient-based evasion attack ($\epsilon = 0.10$) compared to 45.4% during the standard training. Figure 4 verifies that decreasing the false positive rate is achievable by combining GAN with training without any augmentation. Figures 8, 10, and 11 further demonstrate the accuracy and robustness of the proposed method, in both benign and malicious classes, through the ROC-AUC analysis, the confusion-matrix analysis, and the MCC analysis respectively.

The main findings of this work are: (1) A unified architecture for encrypted polymorphic-malware detection; (2) Adversarial augmentation using GANs to enhance accuracy and robustness; (3) A detailed reproducible experimental setup and dataset protocol; (4) A transparent evaluation suite that includes performance, robustness, ROC-AUC, confusion-matrix, and MCC analyses; and (5) Inference latency that is suitable for real-time deployment studies.

Recommendations

- 1) Deploy GANs in NIDS, where the false-positive rate is expensive to operate. For rare polymorphic variants, which are underrepresented, Figure 4 offers quantitative support for the use of augmentation.
- 2) Use explainability interfaces (like GRAD-CAM) along with detection output. The classifier score, influential packet regions and relevant TLS/flow metadata should be displayed concurrently to assist analyst validation.
- 3) Regularly update the CGAN and classifier with new families of malware seen and new encryption behavior. Retraining should be according to drift, incident and labelling capacity.
- 4) When reporting accuracy and F1-score, also report ROC-AUC, confusion matrices, and MCC. These measures exhibit threshold behavior, class-specific errors and balanced classification quality, especially in the presence of class imbalance.

- 5) Benchmark on non-TLS encryption (e.g., WireGuard, QUIC) with the same feature schema, and the same leakage-control and diagnostic-evaluation requirements.
- 6) Investigate federated learning architectures for privacy-preserving training of GANs in a distributed network-monitoring setup, particularly in scenarios where direct sharing of raw traffic is not allowed.

Declarations

Acknowledgments: The author gratefully acknowledges the management of Kogi State Polytechnic, Lokoja, for the institutional support provided for this research.

Artificial Intelligence (AI) Use Statement: The author confirms that no artificial intelligence (AI) tools were used in the conception, analysis, interpretation, or writing of this manuscript.

Author Contribution: The author was solely responsible for the conception and design of the study, implementation of the study protocol, literature review, data collection and curation, model development, statistical analysis, interpretation of results, manuscript preparation, review, editing, revision, and submission of the manuscript.

Conflict of Interest: The author declares that there is no conflict of interest.

Consent to Publish: The author agrees to the publication of this manuscript in International Journal of Recent Innovations in Academic Research.

Data Availability Statement: The datasets used or analyzed during the current study are available from the corresponding author upon reasonable request. The study also uses publicly available benchmark datasets and a custom synthetic polymorphic-malware corpus generated using the proposed CGAN framework.

Funding: This research was funded by the Tertiary Education Trust Fund (TETFund), Nigeria, under the Institution-Based Research (IBR) Intervention.

Institutional Review Board Statement: Not applicable. This study involved computational analysis of encrypted network-traffic data and synthetic data and did not involve human participants, animals, or human-derived biological materials.

Informed Consent Statement: Not applicable. This study did not involve human participants or the collection of personal information directly from human subjects.

Research Content: The author confirms that the research content presented in this manuscript is original and has not been published elsewhere.

References

1. Gazzan, M., Alobaywi, B., Almutairi, M. and Sheldon, F.T. 2025. A deep learning framework for enhanced detection of polymorphic ransomware. *Future Internet*, 17(7): 311.
2. Paul, O. and Moses, T. 2025. Deep learning models for detection and classification of polymorphic malware over encrypted networks consisting of 2D CNN-LSTM, GAN-based adversarial training and GRAD-CAM for polymorphic malware detection explainability. In: *International Conference on Engineering, Science, and Urban Sustainability (ICESUS 2025)* (pp. 307–321). Atlantis Press.
3. Jonath, K.K., Naz, A. and Zhang, S. 2026. GMM-cGAN: Mitigating data scarcity and label noise for robust encrypted malicious traffic classification. *Computer Networks*, 274: 111823.
4. Chopparapu, G. and Kavitha, S. 2025. Exploring the application of generative adversarial networks for encrypted traffic classification in SDN-enabled home networks. *Engineering, Technology and Applied Science Research*, 15(5): 28257–28264.
5. Akpaku, E., Chen, J., Ahmed, M., Leslie Brown-Acquaye, W., Kwadzo Agbenyegah, F. and Nii Ayitey Sosu, R. 2025. Detecting encrypted malicious traffic with HEAT: A header-focused deep learning approach. *The Computer Journal*, 68(12): 2031-2058.
6. Eisa, M., Yardley, Q., Witherspoon, R., Pendlebury, H. and Rutherford, C. 2025. Semantic entanglement-based ransomware detection via probabilistic latent encryption mapping. *arXiv*. <https://arxiv.org/abs/2502.02730>
7. Fortescue, B., Hawksmoor, E., Wetherington, A., Marlowe, F. and Pekepok, K. 2025. Neural encrypted state transduction for ransomware classification: A novel approach using cryptographic flow residuals. *arXiv*. <https://arxiv.org/abs/2502.05341>
8. Gareth, L., Fairbrother, M., Blackwood, P., Underhill, L. and Ruthermore, B. 2025. Intelligent code embedding framework for high-precision ransomware detection via multimodal execution path analysis. *arXiv*. <https://arxiv.org/abs/2501.15836>
9. Mirza, A. and Osindero, S. 2014. Conditional generative adversarial nets. *arXiv*. <https://arxiv.org/abs/1411.1784>

10. Jiang, X., et al. 2024. WGAN-GP for network intrusion detection under class imbalance. In: Proceedings of the 2024 IEEE International Conference on Communications (pp. 1–6).
11. Kumar, M. and Sinha, A. 2023. GAN-based data augmentation for intrusion detection systems. *Journal of Network and Computer Applications*, 215: 103642.
12. Gebrehans, T., et al. 2025. GAN-based malware detection: A comprehensive categorization. *Computers and Security*, 148: 104125.
13. Coppolino, L., et al. 2024. CGAN-based polymorphic malware trace generation. *Future Generation Computer Systems*, 152: 234–248.
14. Chan, P., et al. 2024. GAN-RL hybrids for phishing detection. *IEEE Transactions on Information Forensics and Security*, 19: 2345–2358.
15. Bethu, S. 2024. GAN-based malicious IoT traffic detection. *IEEE Internet of Things Journal*, 11(8): 14567–14578.
16. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. 2014. Generative adversarial nets. In: *Advances in Neural Information Processing Systems* (Vol. 27, pp. 2672–2680). Curran Associates, Inc.
17. Arjovsky, M., Chintala, S. and Bottou, L. 2017. Wasserstein generative adversarial networks. In: *Proceedings of the 34th International Conference on Machine Learning* (pp. 214–223).
18. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A. and Chen, X. 2016. Improved techniques for training GANs. In: *Advances in Neural Information Processing Systems* (Vol. 29, pp. 2234–2242). Curran Associates, Inc.
19. Wang, Z., She, Q. and Ward, T.E. 2024. A novel GAN-based approach for encrypted traffic classification. *IEEE Transactions on Network and Service Management*, 21(3): 2789–2802.
20. Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P., Al-Nemrat, A. and Venkatraman, S. 2019. Deep learning approach for intelligent intrusion detection system. *IEEE Access*, 7: 41525–41550.
21. Xin, Y., Kong, L., Liu, Z., Chen, Y., et al. 2018. Machine learning and deep learning methods for cybersecurity. *IEEE Access*, 6: 35365–35381.
22. Shaukat, K., Luo, S., Varadharajan, V., Hameed, I.A. and Xu, M. 2020. A survey on machine learning techniques for cyber security in the last decade. *IEEE Access*, 8: 222310–222354.
23. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. and Batra, D. 2017. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In: *Proceedings of the IEEE International Conference on Computer Vision* (pp. 618–626).
24. Carlini, N. and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In: *2017 IEEE Symposium on Security and Privacy (SP)* (pp. 39–57). IEEE.
25. Kurakin, A., Goodfellow, I. and Bengio, S. 2016. Adversarial machine learning at scale. *arXiv*. <https://arxiv.org/abs/1611.01236>
26. Goodfellow, I.J., Shlens, J. and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv*. <https://arxiv.org/abs/1412.6572>
27. Madry, A., Makelov, A., Schmidt, L., Tsipras, D. and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv*. <https://arxiv.org/abs/1706.06083>
28. Tramer, F., Carlini, N., Brendel, W. and Madry, A. 2020. On adaptive attacks to adversarial example defenses. *Advances in Neural Information Processing Systems*, 33: 1633–1645.
29. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. and Fergus, R. 2013. Intriguing properties of neural networks. *arXiv*. <https://arxiv.org/abs/1312.6199>
30. Hochreiter, S. and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation*, 9(8): 1735–1780.
31. He, K., Zhang, X., Ren, S. and Sun, J. 2016. Deep residual learning for image recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778). IEEE.
32. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. And Polosukhin, I. 2017. Attention is all you need. In: *Advances in Neural Information Processing Systems* (Vol. 30, pp. 5998–6008).
33. Kingma, D.P. and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv*. <https://arxiv.org/abs/1412.6980>

34. Papernot, N., McDaniel, P., Sinha, A. and Wellman, M. 2016. Towards the science of security and privacy in machine learning. arXiv. <https://arxiv.org/abs/1611.03814>

Citation: Ogba Paul Onu. 2026. Hybrid Deep Learning Framework with GAN-Based Adversarial Training and Explainable AI for Detection of Polymorphic Malware in Encrypted Network Traffic. *International Journal of Recent Innovations in Academic Research*, 10(4): 13-27.

Copyright: ©2026 Ogba Paul Onu. This is an open-access article distributed under the terms of the Creative Commons Attribution International License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.