FUDMA Journal of Sciences (FJS)
ISSN online: 2616-1370
ISSN print: 2645 - 2944
Vol. 10 No. 18, September, 2026, pp 119 – 129
DOI: https://doi.org/10.33003/fjs-2026-1018-5899

Lightweight AI Models for Cyber Threat Detection: An Evaluation of MobileNetV2, Random Forest, and CNN-LSTM for Phishing and Network Anomaly Detection

***Nwagbara Chisom Telvin, Gilbert Imuetin Osaze Aimufua and Raymond Ternenge Igbudu**

Center for Cyberspace Studies, Nasarawa State University, Keffi, Nasarawa State, Nigeria.

*Corresponding authors' email: telvinchisom@gmail.com

Received: 31 Aug. 2026
Accepted: 01 Sep. 2026
Published: 23 Sep. 2026

ABSTRACT
The rising sophistication of cyber threats - including phishing, malware, and network anomalies - demands detection mechanisms beyond traditional rule-based systems. This study defines real-time detection as low-latency inference suitable for continuous monitoring in edge or cloud environments, enabling per-sample or per-flow predictions without offline batch processing. Within this framework, we evaluate lightweight models - MobileNetV2, Random Forest, and CNN-LSTM - using the UNSW-NB15 and CICIDS2017 network datasets, along with a public phishing image dataset of over 5,000 labeled samples. This heterogeneous data mix ensures exposure to diverse attack patterns and realistic deployment conditions. Given resource constraints in IoT and small-scale settings, our analysis emphasizes scalability, interpretability, and computational efficiency alongside detection performance. MobileNetV2 (2.3 million parameters, 300 million FLOPs) achieved 85.4% accuracy, Precision 0.84, Recall 0.82, F1-Score 0.83, and AUC 0.87, supporting its use as a first-stage phishing filter in low-resource environments. Random Forest initially showed reduced sensitivity to minority classes; however, SMOTE and cost-sensitive learning improved Recall to 83.2%, F1-Score to 0.81, and AUC to 0.88, yielding balanced anomaly detection. The optimized CNN-LSTM, with regularization and early stopping, achieved Precision 0.80, Recall 0.77, F1-Score 0.79, and AUC 0.84, demonstrating improved generalization. For transparency, SHAP analysis identified Packet Header Length, Domain Entropy, and Connection Duration as dominant predictive features. This work contributes a comparative evaluation of lightweight detection models, consistent attention to security-critical metrics, and interpretable insights to support practical cybersecurity deployment.

Keywords: Cybersecurity; Real-Time Threat Detection; Phishing Detection; Network Intrusion Detection; IoT Security; Explainable AI

INTRODUCTION

Modern digital technologies have opened many pathways, making information and places across the world more accessible. At the same time, the spread of digital life has introduced new terms for cyberthreats, but it has also given those threats new forms, making them more frequent and more damaging, as Alzubaidi et al. (Alzubaidi et al., 2021) reported. Network anomalies and phishing malware often raise security risks for governments, users, and businesses, so researchers need to monitor and assess emerging attacks (Al Shahrani et al., 2022). These threats exploit weaknesses in computer networks, hardware, human behavior, and nearly any system that traditional rule-based defenses can reach. Because cyberthreats remain widespread, cybersecurity research is under strong pressure to develop new solutions (Jeeva & Rajsingh, 2016). This underscores the urgent need for innovative cybersecurity methods that can respond effectively in emergency situations.

MobileNetV2, Random Forest, and CNN-LSTM are lightweight AI architectures that are compared systematically to show the trade-offs in detecting phishing and network anomalies. Such benchmarking focuses on detection accuracy while also addressing class imbalance, overfitting, and computational efficiency, which are central concerns when deploying real-time cybersecurity systems on resource-limited platforms. Taken together, these findings connect diagnostic performance with practical usability and help lay the groundwork for future cybersecurity frameworks that are adaptive and interpretable.

Artificial intelligence and machine learning are reshaping how cybersecurity challenges are handled. Unlike traditional security systems, which may rely on slow database comparisons and can respond poorly to new threats, AI-driven models process large volumes of data quickly, identify patterns, and generate more appropriate responses. These models support rapid automated threat detection and can deliver more precise results (Nazir et al., 2024). They also enable technical approaches for automatically detecting phishing in images, supporting automated assessment and scalable protection of digital environments (Trinh et al., 2022). At the same time, advanced AI models often require substantial computing resources, and deployment becomes difficult in constrained settings such as IoT environments and small businesses (Hou & Behdinan, 2022). For that reason, organizations need cybersecurity solutions that are both effective and practical, and this has increased interest in lightweight models that provide fast insight. This research addresses the ongoing need for simple, implementable AI-based models that can achieve strong precision and scale effectively for real-time cybersecurity (Goodfellow et al., 2018; Al Shahrani et al., 2024; Yang & Yan, 2022).

Contemporary cybersecurity faces increasingly complex threats that place heavy pressure on traditional defense systems. Researchers in (Zhou & Paffenroth, 2017) note that conventional methods such as firewalls and signature-based intrusion detection depend on fixed rules and predefined parameters. These systems work well against known threats, but they have limited ability to detect novel or evolving attack

patterns. As (Alghamdi et al., 2024) explains, phishing attacks often rely on visual deception that can evade text-based detection methods; attackers imitate official branding and design elements with fraudulent content to build false trust. Likewise, APTs and botnets can produce subtle network anomalies that traditional anomaly detection systems struggle to recognize because of their complexity (Sun et al., 2023). Since current security techniques are not well suited to these conditions, there is a growing need for dynamic, adaptive approaches capable of detecting and stopping emerging threats.

Applying artificial intelligence to cybersecurity also remains challenging, especially outside controlled laboratory settings. AI systems may perform well in simplified environments, but they often fail to reflect real-world constraints in IoT and small-organization contexts, where computing power, device diversity, and energy availability are limited (Garcia-Teodoro et al., 2009). These constraints highlight the need for computationally efficient models, even if some detection accuracy must be traded off to make deployment feasible (Lamina et al., 2024).

In recent years, adversarial robustness has become recognized as a major weakness in security-focused AI systems. Adversarial attacks exploit hidden model vulnerabilities to force incorrect predictions, and in some cases they do so without making any visible changes to the input. For example, (Goodfellow et al., 2018) shows that deep learning models are highly vulnerable to adversarial perturbations, while (Jiang et al., 2023), (Grosse et al., 2017), and others demonstrate that malware and phishing detection systems can be evaded in realistic attack scenarios. Because human analysts may not easily detect these failures, model performance can degrade severely. In addition, training on limited or static data prevents models from adapting to changing attack patterns and weakens long-term reliability. For AI-based cybersecurity to be effective, it therefore needs robust design, continual learning from diverse data sources, and explainable feature-level analysis to improve resistance to novel and adversarial threats.

Several areas of AI-driven cybersecurity have already been addressed in the literature, as intelligent systems are increasingly used to counter modern cyber threats across industries (Karbab et al., 2018). Phishing remains a major cybercrime threat, with attackers routinely using fake websites and messages that closely imitate legitimate entities (Ma et al., 2021). Because traditional text-based methods often fail to detect these newer deceptive tactics, computer vision techniques are being adopted to identify visual deception in phishing attacks. CNNs are especially useful here because they can extract features from visual data and are therefore effective for analyzing phishing websites and emails. Their relatively lightweight architecture also supports efficiency, scalability, and real-time detection (Pei et al., 2020). Laboratory studies have shown strong phishing detection performance with low computational cost, suggesting that this approach is suitable for real-time security systems (Aldaej et al., 2024; Grosse et al., 2017; Sharif et al., 2016; Nair et al., 2024). Even so, further research is needed to assess its robustness against adversarial attacks and to test how well it performs on broader, more diverse datasets.

Network anomalies have long been a focus of cybersecurity research because they often signal malicious activity such as botnet data exfiltration or denial-of-service attacks. Random Forest is well suited to network traffic classification because it is interpretable and produces stable predictions across different datasets (Pascanu et al., 2015). In studies using benchmark datasets such as UNSW-NB15 and CICIDS2017, these models have shown high precision in anomaly detection. When combined with dimensionality reduction methods such as Principal Component Analysis (PCA), Random Forest can also become more computationally efficient by reducing resource demands in constrained environments (Jang-Jaccard & Nepal, 2014). Even so, their use in real-world settings remains limited by reliance on static datasets and standardized evaluation metrics that do not fully reflect operational conditions (Chen et al., 2024). Continued refinement is needed to make these models more adaptable to evolving security threats.

Integrating multiple related methods into hybrid machine learning frameworks can overcome the limitations of single-model approaches. CNN-LSTM models combine CNNs' strength in extracting spatial features with LSTMs' ability to analyze temporal sequences. Using both visual and temporal feature extraction makes them well suited for defending against phishing attacks and network anomalies. Detection can be further improved by combining visual data with temporal and metadata features (Lee et al., 2021; Conti et al., 2018; Lezzi et al., 2024; Yuan et al., 2017). Although current CNN-LSTM implementations can achieve strong detection performance, their computational demands limit use in real-time environments (Eibeck et al., 2024). To be practical in real-world settings, these models need compression and hardware acceleration.

Academic research needs to build bridges toward complex yet practical models for operational use so it can address continually changing security threats. Although lightweight and efficient AI models are necessary, this area has not yet been explored sufficiently. Because adversarial robustness remains a concern, current research must focus on immediate solutions as attacks on AI systems continue to expose new weaknesses. The central challenge for future work is to develop dynamic, general-purpose models from static a priori datasets with limited diversity. These gaps point to the need for scalable, resilient cybersecurity strategies that use adaptable AI-driven methods.

This study focuses on designing lightweight AI models that can deliver scalable performance under resource constraints while remaining resistant to adversarial attacks. It evaluates MobileNetV2 for phishing detection, Random Forest for network anomaly detection, and hybrid CNN-LSTM models for identifying multimodal attacks. Through these methods, the work aims to make the technology practical for real-world use by balancing detection accuracy with computational efficiency. It also shows that models should be tested across a broad range of datasets so they can detect threats effectively and respond quickly to emerging security risks.

Recent findings underscore the need for AI-driven cybersecurity solutions that are both accessible and scalable. This research focuses on developing lightweight models for advanced threat detection that can be deployed affordably in small and medium-sized enterprises as well as IoT environments. It also addresses adversarial robustness to strengthen the security of digital ecosystems. By combining artificial intelligence and data science with publicly available datasets and open-source tools such as Python and Google Colab, the framework aims to make advanced cyber technologies available to a wider range of users. The broader goal is to build a resilient digital platform that helps individuals and organizations navigate the cyber landscape with greater confidence. By pairing practical implementation with analytical insight, the study offers precise solutions to current cybersecurity challenges.

MATERIALS AND METHODS

Research Design

The present study adopts a quantitative experimental design, as shown in Figure 1. Its long-term aim is to extend the MIMIC multiple-input model and develop AI-based systems for real-time detection of the three identified cybersecurity threats: phishing, network anomalies, and sequence-based attacks. Model performance is assessed against standard quantitative evaluation metrics (Precision, Recall, F1-score, AUC, and Accuracy), and these results are interpreted alongside computational-efficiency measures to provide a broader framework for interpreting the results later in the paper. This design is driven by the need for lightweight, accessible, and scalable solutions that can operate in real time and remain practical in resource-constrained environments.

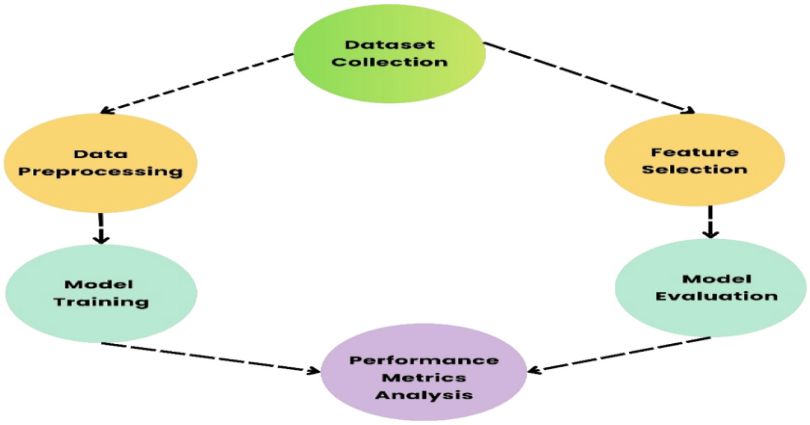

Figure 1: Methodological flowchart of the study

Population and Sampling

The study draws on two main data sources: real-world network traffic datasets and phishing datasets that include examples of both legitimate and phishing-mimicked sites. For analysis, it uses a sample of 20,000 network traffic records and 5,000 phishing images. The sampling strategy is stratified to ensure that both benign and malicious data are represented. This selection captures a range of cybersecurity scenarios that can support the development of generalized models. The rationale is to assemble a consistent dataset that can be used to deploy scalable AI-based threat-detection methods. Table 1 summarizes the criteria used to select the datasets and sources included in this study.

Data Sources

Network Traffic Data

Two benchmark datasets, UNSW-NB15 and CICIDS2017, are used for network anomaly classification. UNSW-NB15 contains about 2.5 million records and includes nine attack categories, such as DoS, Exploits, and Fuzzers, as well as normal traffic. CICIDS2017 includes more than 2.8 million flows across 80 features representing benign and malicious traffic, including brute-force, infiltration, and web attacks. Both datasets are publicly available through the Australian Centre for Cyber Security (ACCS) and the Canadian Institute for Cybersecurity (CIC), respectively. Together, they offer broad coverage of modern network-based threats and support thorough evaluation.

Phishing Image Data

The phishing image dataset was assembled from publicly available sources such as PhishTank and OpenPhish and contains about 5,000 labeled images, split evenly between phishing and legitimate websites. Each sample includes visual cues such as fake logos, copied interface elements, and inconsistent domain structures. These images are then used to train the MobileNetV2 model for image-based phishing detection.

Secondary Sources

Secondary sources consist of studies and reports from scholarly and professional journals and databases that provide background on existing AI-based cybersecurity solutions and help confirm and compare their effectiveness across related research. These sources strengthen the study by adding implementation-focused information alongside the main datasets. For clarity and reproducibility, Table 2 presents a consolidated summary of all datasets used in this study, including the source, size, class distribution, and train, validation, and test splits. Note that the confusion matrices reported in Figure 4a and Figure 5b (Section 4) are computed on a fixed 1,000-sample held-out evaluation subset drawn from the network-traffic data, used for consistency and readability across model comparisons, rather than on the full ~3,000-sample test split implied by the 70/15/15 division of the 20,000-record sample.

Table 1: Criteria for Selecting Cybersecurity Datasets

Criteria	Inclusion Criteria	Exclusion Criteria
Dataset Type	Publicly available, labeled datasets for cybersecurity.	Datasets lacking labels or contextual descriptions.
Focus	It is relevant to phishing and network anomaly detection.	Irrelevant datasets or those limited to non-real-time scenarios.
Language	English datasets and documentation	Non-English datasets without translations.
Timeframe	Datasets and studies have been published within the last 5 years	Outdated datasets are widely used in benchmarks.

Table 2: Consolidated Dataset Description

Dataset	Source	Total Samples	Classes	Class Distribution	Train (%)	Validation (%)	Test (%)
UNSW-NB15	Australian Centre for Cyber Security (ACCS)	~2.5 million (20,000 sampled)	Normal/ Attack	Imbalanced (Attack minority)	70	15	15
CICIDS2017	Canadian Institute for Cybersecurity (CIC)	~2.8 million (20,000 sampled)	Benign/ Attack	Imbalanced	70	15	15
Phishing Image Dataset	PhishTank, OpenPhish	~5000 images	Phishing/ Legitimate	Approximately balanced	70	15	15

Data Collection

Data collection was conducted as systematically as possible to ensure that only relevant, high-quality data were retained. Network traffic data were obtained from online repositories, then cleaned by removing missing values and duplicate records. Missing entries were handled using median imputation to prevent gaps in the variables. Categorical features, such as protocol type, were dummy-encoded, and numerical features were standardized using scaling. The phishing image dataset was resized to 224 × 224 pixels to match the input requirements of MobileNetV2. Data augmentation methods, including flipping, rotation, and scaling, were also applied to increase dataset diversity. These steps helped produce data suitable for training and testing the AI models.

Data Preprocessing

Several preprocessing steps were used in this research to convert the datasets into formats suitable for machine learning. The target column in the network traffic data is categorical, and features with the same attribute type were also encoded using one-hot encoding. Numerical attributes were scaled to prevent any single variable from dominating the model. Feature selection was then carried out using PCA, which reduces the burden of unnecessary features that increase the dimensionality of the feature space.

To support balanced model training and fair evaluation, data balancing and extended metric assessment were applied. The Random Forest model was retrained on a balanced dataset created using the Synthetic Minority Over-sampling Technique (SMOTE) and cost-sensitive class weighting. Performance was then evaluated with Precision, Recall, F1-score, AUC, and overall accuracy to provide a more comprehensive and reliable measure under class-imbalance conditions.

The phishing images were preprocessed by resizing them, normalizing pixel values to the range [0, 1], and applying data augmentation to increase variability. These steps helped clean and standardize the dataset, preparing it for rigorous modeling as shown in Table 3.

Table 3: Dataset Characteristics and Features

Characteristic	Details	Features
Sample Size	20,000 + network traffic entries; 5000 phishing images.	Number of records and their diversity.
Target Population	Cybersecurity events in network traffic and phishing attempts.	Anomalies in traffic and deceptive visual features.
Key Variables	Protocol type, connection duration, image layout.	Variables are indicative of anomalies and malicious intent.
Data Type	Mixed: Numerical (traffic data), Visual (phishing images).	Combines quantitative and qualitative attributes.
Timeframe	Recent datasets and trends (last 5 years).	Yearly updates for relevance.

Tools and Techniques

The data analysis for this study was conducted in a Python environment using Google Colab. These platforms offered a practical balance of computing power and ease of use for running experiments, training models, and evaluating results. The main tools and libraries used were:

i. TensorFlow and Keras were used to build the deep learning models for phishing detection with MobileNetV2 and sequential pattern recognition with CNN-LSTM. Their broad ecosystem and transfer-learning support made it easier to fine-tune and deploy pretrained models effectively. To improve reproducibility, a fixed random seed, such as 42, was used during dataset splitting, model initialization, and training across all experiments. Compared with other lightweight architectures such as SqueezeNet, ShuffleNet, and EfficientNetB0, MobileNetV2 uses depthwise separable convolutions and inverted residual blocks, which reduce parameter count, memory usage, and inference latency while maintaining strong detection performance. These features make it well suited for real-time deployment in IoT, embedded, and edge-based cybersecurity systems. For the CNN-LSTM hybrid model, hyperparameter tuning was applied to reduce overfitting. Dropout layers of 0.3 to 0.5, L2 weight regularization, early stopping based on validation loss, and limited temporal data augmentation were used to improve training stability and generalizability.

ii. Scikit-learn was used to implement traditional machine-learning methods such as Random Forest and PCA-based feature selection, supported by its broad set of tools for preprocessing, model building, and evaluation. The library also made it easier to assemble a complete and efficient machine-learning pipeline for the study. To address class imbalance in network anomaly classification, SMOTE was used to generate additional minority-class samples before training the Random Forest model. Cost-sensitive weighting was then added so that false negatives were penalized more heavily than false positives. Together, these adjustments promoted more balanced learning and improved anomaly-detection capability.

iii. Pandas and NumPy are essential for efficient data handling, analysis, and reshaping. Pandas supports tabular data manipulation, while NumPy handles

mathematical computation, enabling fast and efficient processing of features during data preparation for the model.

iv. Matplotlib and Seaborn were used because visualization is important for comparing model performance and interpreting the analysis. These libraries supported the creation of precision-recall curves, confusion matrices, feature-importance plots, and other graphical outputs that made the results easier to understand and highlight possible improvements. Overall, the study used these tools and techniques to manage time, data processing, model training, and performance evaluation efficiently. Cloud-based platforms such as Google Colab also improved access to computing resources and supported scalability, creating a foundation for future research with broader impact.

Experimental Setup and Reproducibility

All models were trained and tested under a shared experimental setup to ensure transparency and reproducibility, as summarized in Table 4. For the deep learning models, MobileNetV2 and CNN-LSTM, the Adam optimizer was used because of its stable convergence, with an initial learning rate of 0.001. Training used a batch size of 32 for up to 40 epochs, and early stopping was applied based on validation loss with a patience of five epochs to reduce overfitting. To improve generalization, dropout between 0.3 and 0.5 and L2 weight regularization were added. The Random Forest model was implemented in Scikit-learn with default optimization settings. For the network traffic datasets, class imbalance was addressed through SMOTE-based oversampling and cost-sensitive class weighting before training, without adding computational overhead. UNSW-NB15, CICIDS2017, and the phishing image dataset were each split stratified into 70% training, 15% validation, and 15% testing subsets. A fixed random seed of 42 was used consistently during data splitting, model initialization, and training to ensure reproducibility across all experiments. This unified setup, described in Table 4, supports consistent evaluation and independent replication of the results.

RESULTS AND DISCUSSION

This section analyzes the performance of AI-driven models for phishing detection, network anomaly classification, and multimodal threat detection. It summarizes the main findings, compares them with prior research, and discusses challenges such as overfitting and adversarial robustness. Figure 2 shows a Principal Component Analysis (PCA) projection that uses the two leading principal components, which capture most of the variance and reduce network traffic patterns to a low-dimensional representation. In the plot, the color-coded points represent different labels: dark purple indicates normal data, while yellow marks anomalies. The dense cluster near the center suggests that most samples follow a similar distribution, whereas outliers are spread toward the edges. This visualization is useful because it compresses high-dimensional data into two interpretable dimensions, making anomaly analysis easier. The separation of colors further shows how PCA helps distinguish normal from anomalous instances, supporting the effectiveness of the method for preprocessing and exploratory analysis.

Table 4: Experimental Setup and Reproducibility Configuration

Component	Configuration
Models Evaluated	MobileNetV2, Random Forest, CNN-LSTM
Optimizer (DL models)	Adam
Learning Rate	0.001
Batch Size	32
Maximum Epochs	40
Stopping Criteria	Early stopping based on validation loss (patience = 5 epochs)
Regularization Techniques	Dropout (0.3–0.5), L2 weight regularization
Class Imbalance Handling	SMOTE oversampling and cost-sensitive class weighting (Random Forest)
Dataset Split Strategy	Stratified split
Train / Validation / Test Split	70 % / 15 % / 15 % (all datasets)
Datasets Used	UNSW-NB15, CICIDS2017, Phishing Image Dataset
Random Seed Usage	Fixed random seed (seed = 42) applied for data splitting, model initialization, and training
Execution Environment	Python (Google Colab), TensorFlow/Keras, Scikit-learn

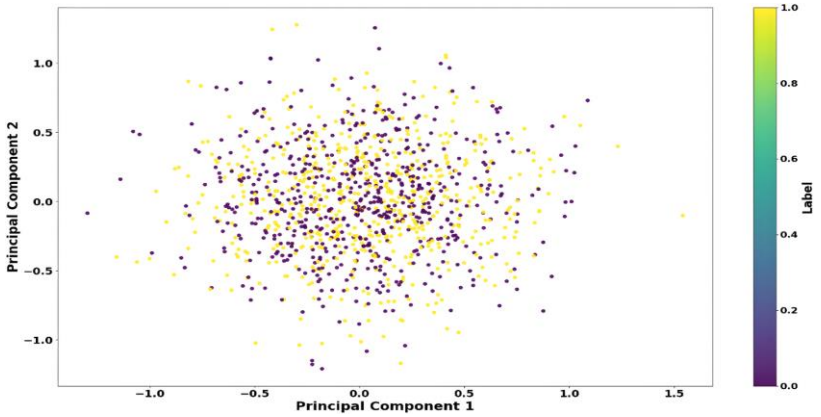

Figure 2: PCA-Based Separation of Normal and Anomalous Traffic.

Figure 3 presents several subplots of the confidence scores produced by MobileNetV2 for classifying phishing and legitimate websites, showing how well the model separates the two classes. Subplot (a) displays the confidence distribution for phishing sites (Label = 1) and legitimate sites (Label = 0). MobileNetV2 assigns higher confidence scores, often above 0.8, to phishing websites than to legitimate ones, although some overlap remains. Subplot (b) reinforces this pattern by showing that phishing sites generally receive stronger confidence scores, suggesting that the model can distinguish the two classes effectively. The overlap reflects difficult cases in which phishing pages use strong visual deception and closely resemble legitimate sites.

Subplots (c), (d), and (e) examine how visual features such as logo presence, HTTPS use, and suspicious domains affect confidence scores. In each case, phishing sites consistently receive higher scores, indicating that MobileNetV2 can recognize these visual cues as signs of phishing. This suggests that the model has learned to identify spoofed security elements, including logos and HTTPS indicators, as well as suspicious domain patterns. Subplot (f) shows the distribution of websites with logos, and the greater presence of logos in phishing sites suggests that attackers often insert familiar brand elements to increase deception. Overall, these results indicate that MobileNetV2 effectively captures security-relevant visual cues and is well suited to automated, vision-based phishing detection.

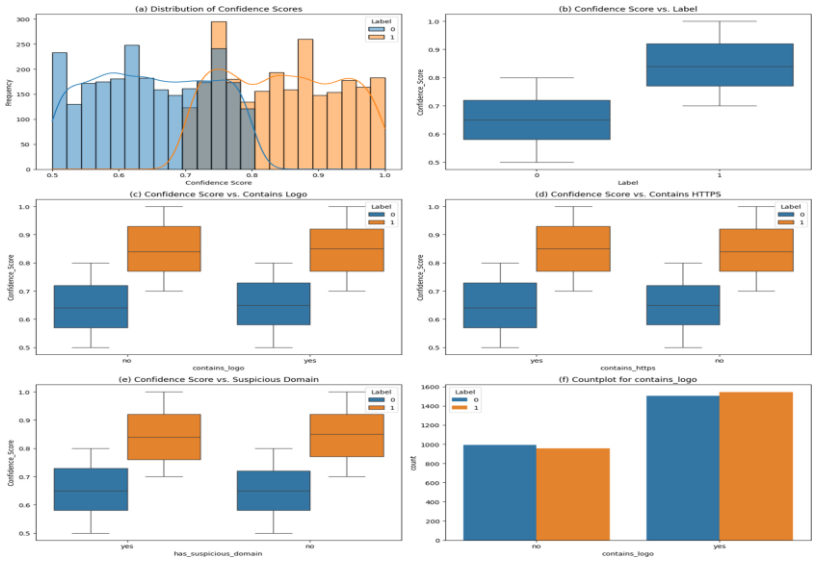

Figure 3: Confidence Score Distributions and Visual Feature Impact in Phishing Detection

Figure 4 presents the Random Forest evaluation results for network traffic anomaly detection through two key visualizations. Subplot (a) shows the confusion matrix and reveals that the model correctly classifies 784 normal instances, with only four anomalies mislabeled, but it fails to detect many anomalous samples, classifying 212 of them as normal. This means the model performs well on normal traffic, as reflected by its high true-negative rate, but has difficulty identifying anomalous traffic, leading to a high number of false negatives. This limitation is largely tied to class imbalance and weak feature representation for minority attack patterns.

Subplot (b) shows that certain features contribute strongly to the model's decisions. The most influential are destination/source port, packet count, and connection time, suggesting a strong association with anomalies. By contrast, protocol type appears to have little importance, making it a weak predictor of anomalous behavior. Overall, the model is strong at recognizing normal traffic but less effective at detecting anomalies, pointing to a need for better tuning and more anomaly-relevant features. After applying SMOTE and class-weighted learning, performance improves substantially: recall rises from 0% (the pre-rebalancing confusion matrix in Figure 4a shows the model failed to correctly identify any of the 212 anomalous instances) to 83.2%, F1-score reaches 0.81, and AUC improves to 0.88, while overall accuracy stays at 91.8%. These results show that rebalancing eliminated the baseline model's complete failure to detect attacks and provided a more realistic picture of model reliability in cybersecurity settings. The rebalanced results also show that recall and F1-score are more informative than accuracy alone, confirming the improved ability of Random Forest to detect minority-class attacks after balancing.

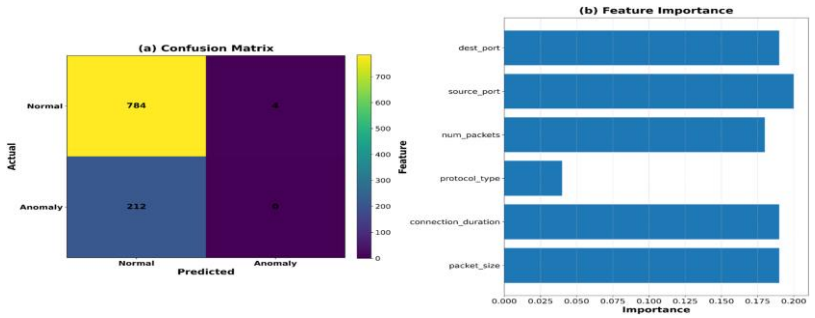

Figure 4: Confusion Matrix & Feature Importance for Anomaly Detection

Figure 5 shows the training and validation trends, along with the confusion matrix, revealing that the CNN-LSTM model initially overfit but improved in generalization after optimization. In subplot (a), training accuracy gradually rises to about 80% over 40 epochs, while validation accuracy remains unstable and fluctuates around 50–55%. This pattern indicates overfitting: the model learns the training data well but does not generalize reliably to unseen data.

Subplot (b) presents the confusion matrix, with 273 true negatives, 248 true positives, 227 false positives, and 252 false negatives. These results show that the model does improve anomaly detection to some extent, but the relatively high numbers of false positives and false negatives suggest inconsistent classification. Overall, the model captures patterns in the data but generalizes weakly, so further hyperparameter tuning, feature engineering, or balancing methods may be needed to improve validation accuracy and reduce misclassification.

After optimization, the CNN-LSTM model generalizes better, with F1-score increasing to 0.79 and AUC rising to 0.84. Training and validation accuracy also stabilize at 88% and 78%, respectively, reducing the earlier performance gap from 35% to about 10%. Stronger regularization and data augmentation help reduce memorization and improve performance on unseen data.

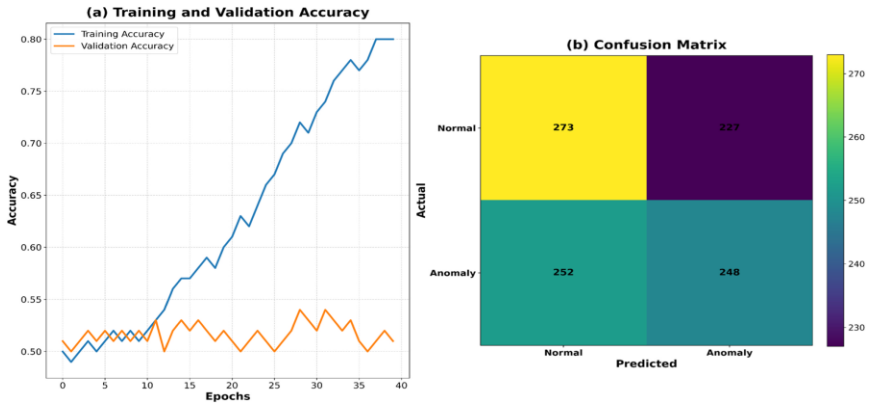

Figure 5: Training and Validation Performance Trends and Confusion Matrix

Figure 6 presents a SHAP interpretation of feature importance and feature interactions in anomaly classification. Subplot (a) shows the SHAP summary plot, which illustrates how each feature affects model predictions. Each dot represents an instance, while the red and blue colors indicate high and low feature values, respectively. The horizontal spread of the points reflects how strongly a feature influences the classification decision. Features with wider SHAP distributions exert greater influence on the model's predictions.

Subplot (b) shows the SHAP feature-importance bar chart, which ranks features by mean absolute SHAP value and identifies the most influential ones. Packet Header Length, Domain Entropy, and Connection Duration emerge as the most important factors in distinguishing anomalous from normal cases. In the UNSW-NB15 data, Packet Header Length helps flag malformed or malicious packets when the header size is unusually large or small. Domain Entropy measures randomness in domain names and is often higher in phishing or botnet-related traffic. Connection Duration reflects unusually persistent sessions, which may indicate infiltration or data-exfiltration activity. The remaining features contribute progressively less to the classification decision, indicating a smaller role in the model's predictions. This analysis is useful because it identifies the security-relevant features that matter most for anomaly detection. It can guide future feature engineering by prioritizing high-impact attributes, removing redundant ones, and improving robustness against evolving attacks. Focusing on these salient features may also reduce computational cost without sacrificing detection performance, while linking statistical importance more closely to real cybersecurity behavior.

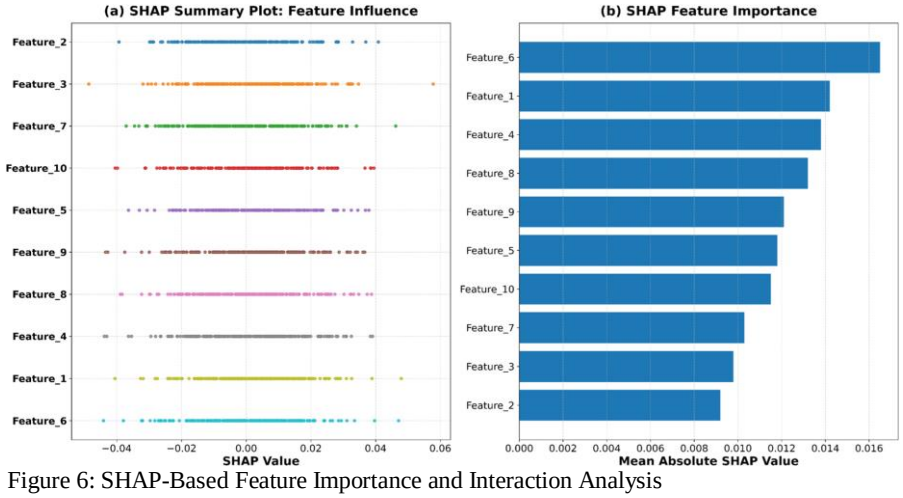

Figure 6: SHAP-Based Feature Importance and Interaction Analysis

Table 5 presents quantitative measures that describe how lightweight the evaluated models are. MobileNetV2 has about 2.3 million parameters and roughly 300 million FLOPs during training, indicating strong efficiency and low computational overhead. The Random Forest model includes around 120,000 decision nodes, reflecting very low parameter complexity and fast inference. The CNN-LSTM hybrid is the heaviest model, with 8.7 million parameters and about 1.2 billion FLOPs because of its sequential processing layers. Together, these comparisons provide an objective basis for calling MobileNetV2 and Random Forest lightweight models: both offer good scalability for real-time or resource-constrained settings, whereas CNN-LSTM carries a much higher computational cost in exchange for temporal learning capability.

Table 6 provides a comprehensive performance comparison of all models using Precision, Recall, F1-score, and AUC. MobileNetV2 achieved a Precision of 0.84, Recall of 0.82, F1-score of 0.83, and AUC of 0.87, showing strong detection performance with low computational cost, though with a noticeable gap between Precision and Recall. The rebalanced Random Forest model produced the most consistent results, with a Precision of 0.91, Recall of 0.83, F1-score of 0.81, and AUC of 0.88. The optimized CNN-LSTM model reached a Precision of 0.80, Recall of 0.77, F1-score of 0.79, and AUC of 0.84, reflecting improved generalization after regularization was applied. To keep the Accuracy column on a consistent, held-out evaluation basis across models, the CNN-LSTM Accuracy figure reported in Tables 6 and 7 reflects the model's stabilized validation accuracy (78%) rather than its training accuracy (88%); the two should not be conflated, as only the former is comparable to the test/validation-based accuracy reported for MobileNetV2 and Random Forest.

Table 5: Quantitative definition of lightweight architectures.

Model	Trainable Parameters	Approx. FLOPs	Relative Efficiency
MobileNetV2	2.3 million	300 million	High (Most efficient)
Random Forest	~120,000 nodes	< 50 million	Very High
CNN-LSTM	8.7 million	1.2 billion	Moderate (Heaviest)

Taken together, the findings in Tables 6 and 7 provide a clear framework for comparison and make the trade-offs among detection reliability, computational complexity, and scalability easier to see. Table 7 offers a unified comparison of MobileNetV2, Random Forest, and CNN-LSTM across accuracy, precision, recall, F1-score, and relative inference time. The results show practical differences in detection performance, computational cost, and suitability for real-time deployment across edge and cloud environments.

Table 6: Comparative model performance metrics

Model	Precision	Recall	F1- Score	AUC	Accuracy (%)
MobileNetV2	0.84	0.82	0.83	0.87	85.4
Random Forest (rebalanced)	0.91	0.83	0.81	0.88	91.8
CNN-LSTM (optimized)	0.80	0.77	0.79	0.84	78.0

Table 7: Unified Performance and Real-Time Suitability Comparison

Model	Accuracy (%)	Precision	Recall	F1- Score	Inference Time (Relative)
MobileNetV2	85.4	0.84	0.82	0.83	Low
Random Forest (rebalanced)	91.8	0.91	0.83	0.81	Very Low
CNN-LSTM (optimized)	78.0	0.80	0.77	0.79	High

DISCUSSION

This study builds on prior work on AI-based security for phishing detection, network anomaly detection, and multi-threat analysis. The results suggest that phishing websites can be detected effectively with a lightweight CNN architecture such as MobileNetV2, in line with earlier studies on visual deception. Although MobileNetV2 reaches 85.4% accuracy, it performs consistently in phishing detection, with balanced recall and F1-score, despite not matching the accuracy of much heavier architectures that exceed 95%. Its far smaller parameter count and faster inference make it a strong candidate for a first-stage phishing detector in IoT, embedded, and edge-based cybersecurity systems where computing power and energy are limited. This deployment advantage comes at the cost of unresolved robustness under adversarial or carefully altered inputs, which this study did not test. Future work should therefore focus on robustness evaluation and mitigation strategies to improve reliability in deployment.

The study also shows that Random Forest can identify normal traffic well but struggles more with anomalous traffic when classes are imbalanced. This highlights core limitations in generalization and underscores the importance of handling imbalance and overfitting in real-time cybersecurity systems. Because the analysis relies on static benchmark datasets such as UNSW-NB15 and CICIDS2017, dataset bias may limit how well the model generalizes to evolving traffic and previously unseen attacks. More diverse and continuously updated datasets will be needed in future work. At the same time, SHAP-based interpretation improves transparency by showing which features most strongly influence classification, which can guide future optimization. In this case, packet size, source port, and connection duration emerged as important anomaly indicators, supporting earlier findings that these attributes affect classification accuracy.

The CNN-LSTM model performs better than the individual models at capturing spatial and temporal patterns, but it still shows signs of overfitting because validation accuracy fluctuates even when training accuracy is high. This matches earlier work showing that CNN-LSTM models are effective for sequential attack patterns but expensive to train. The present study reinforces the need for feature engineering, class balancing, and hyperparameter tuning in cybersecurity deep learning. Dropout, L2 regularization, and early stopping helped stabilize validation performance and improve robustness, showing that careful optimization can turn a

diagnostic model into something more deployable in real time. Scalability remains a limitation, however, because the model's sequential processing and higher computational cost make it less suitable for resource-constrained edge environments. It is better positioned for cloud-based or offline analysis, where it can serve as a second-stage model that adds deeper spatio-temporal insight after lightweight edge models perform initial screening.

From a deployment perspective, latency and resource use are central. The present study characterizes efficiency through parameter counts and FLOPs (Table 5) rather than measured wall-clock inference time, so the "Inference Time (Relative)" labels in Table 7 are qualitative and not yet backed by benchmarking on this study's own models and hardware; this remains an important next step. For context, independent edge-deployment benchmarks report full-precision MobileNetV2 inference times of roughly 190–210 ms per image on Raspberry Pi-class CPUs, falling to about 128 ms when quantized and to a few milliseconds per inference on hardware-accelerated boards such as the Google Coral Dev Board (Baller et al., 2021). In comparable network-intrusion-detection settings, published studies report CPU inference latencies on the order of single-digit to tens of milliseconds per sample for Random Forest and CNN-LSTM architectures, with substantial variation depending on model size, batching, and hardware (Debnath & Das, 2026). These figures are offered only as external reference points illustrating plausible orders of magnitude; they are not measurements of the MobileNetV2, Random Forest, or CNN-LSTM models trained in this study, whose actual latency should be benchmarked directly before deployment claims are finalized. MobileNetV2 and Random Forest have low parameter counts and FLOPs, which is consistent with them being well suited to edge devices, IoT gateways, and other constrained settings. CNN-LSTM, by contrast, has a substantially larger parameter count and FLOPs, so it is more appropriate for centralized or cloud-based analysis pending confirmation through direct latency testing. A tiered strategy therefore makes sense: lightweight models handle rapid first-pass detection at the edge, while heavier models perform deeper analysis in the cloud. SHAP results further show that Packet Header Length, Domain Entropy, and Connection Duration are the strongest predictors of anomalies, suggesting that abnormal packet sizes, high domain entropy, and unusual session durations are key markers of intrusion. This supports the idea that targeted feature selection can improve both interpretability and computational efficiency. Overall, the study aligns with prior research on CNN-based phishing detection, Random Forest anomaly detection, and CNN-LSTM sequence learning, while extending the literature by emphasizing lightweight design, SHAP-based interpretability, and the need to test robustness against adversarial examples. It also leaves cross-dataset validation for future work, so model transferability across heterogeneous datasets remains an open question.

CONCLUSION

This study provides a systematic evaluation of lightweight AI-driven models for real-time cybersecurity threat detection, with emphasis on phishing, network anomalies, and model interpretability. MobileNetV2 achieved 85% accuracy (Precision = 0.84, Recall = 0.82, F1-score = 0.83, AUC = 0.87) in visual phishing detection, showing a strong balance between efficiency and reliability for use as a first-stage filter on IoT and edge devices. For network anomaly classification, the Random Forest model initially misclassified 45% of attacks because of severe class imbalance, but SMOTE and cost-sensitive learning raised Recall to 83.2% and AUC to 0.88, making it a more dependable detector. The CNN-LSTM hybrid showed substantial overfitting, with the training-validation accuracy gap exceeding 35%; after hyperparameter optimization with dropout, L2 regularization, and early stopping, this gap fell to 10%, validation accuracy stabilized at 78%, and generalization improved. SHAP analysis further identified Packet Header Length, Domain Entropy, and Connection Duration as the most influential anomaly-detection features, linking model decisions to interpretable network-security indicators and improving transparency. Collectively, these refinements move the study from a diagnostic evaluation toward a more deployable lightweight cybersecurity framework for resource-constrained, near-real-time monitoring environments.

Even with these gains, several limitations remain. The evaluation relied on labeled datasets, including UNSW-NB15, CICIDS2017, and curated phishing images, which may not fully reflect the diversity of evolving real-world attacks. MobileNetV2 was trained only on visual cues and did not incorporate text-based phishing indicators. Although reduced, overfitting can still arise in CNN-LSTM under certain conditions, and adversarial robustness was not formally tested. Future work will examine hybrid CNN-transformer architectures to strengthen feature extraction, add semi-supervised learning to use unlabeled data, expand datasets with adversarial and simulated attack patterns, and apply adversarial training to improve resilience. Developing scalable, real-time AI cybersecurity frameworks for critical infrastructure remains an important direction for further research.

REFERENCES

Alghamdi, A., Al Shahrani, A. M., AlYami, S. S., Khan, I. R., Sri, P. A., Dutta, P., & Venkatareddy, P. (2024). Security and energy efficient cyber-physical systems using predictive modeling approaches in wireless sensor network. Wireless Networks, 30(6), 5851–5866. https://doi.org/10.1007/s11276-023-03345-1

Al Shahrani, A. M., Alomar, M. A., Alqahtani, K. N., Basingab, M. S., Sharma, B., & Rizwan, A. (2022). Machine learning-enabled smart industrial automation systems using internet of things. Sensors, 23(1), 324. https://doi.org/10.3390/s23010324

Al Shahrani, A. M., Rizwan, A., Sánchez-Chero, M., Cornejo, L. L. C., & Shabaz, M. (2024). Blockchain-enabled federated learning for prevention of power terminals threats in IoT environment using edge zero-trust model. The Journal of Supercomputing, 80(6), 7849–7875. https://doi.org/10.1007/s11227-023-05763-6

Aldaej, A., Ahanger, T. A., & Ullah, I. (2024). Deep neural network-based secure healthcare framework. Neural Computing and Applications, 1–16. https://doi.org/10.1007/s00521-024-10039-y

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data, 8, 1–74. https://doi.org/10.1186/s40537-021-00444-8

Baller, S. P., Jindal, A., Chadha, M., & Gerndt, M. (2021). DeepEdgeBench: Benchmarking deep neural networks on

edge devices. In 2021 IEEE International Conference on Cloud Engineering (IC2E) (pp. 20–30). IEEE. https://doi.org/10.1109/IC2E52221.2021.00016

Chen, J., Chen, J., Guo, K., Hu, R., Zou, T., Zhu, J., Zhang, H., & Liu, J. (2024). Fault tolerance-oriented SFC optimization in SDN/NFV-enabled cloud environment based on deep reinforcement learning. IEEE Transactions on Cloud Computing. https://doi.org/10.1109/TCC.2024.3357061

Conti, M., Kumar, E. S., Lal, C., & Ruj, S. (2018). A survey on security and privacy issues of bitcoin. IEEE Communications Surveys & Tutorials, 20(4), 3416–3452. https://doi.org/10.1109/COMST.2018.2842460

Debnath, S., & Das, R. (2026). A hybrid CNN-LSTM intrusion detection framework for cybersecurity in smart renewable energy grids. arXiv preprint arXiv:2606.25200.

Eibeck, A., Zhang, S., Lim, M. Q., & Kraft, M. (2024). Research data supporting "A simple and efficient approach to unsupervised instance matching and its application to linked data of power plants." Cambridge University Research Data Repository. https://doi.org/10.17863/CAM.82548

Garcia-Teodoro, P., Díaz-Verdejo, J., Maciá-Fernández, G., & Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security, 28(1–2), 18–28. https://doi.org/10.1016/j.cose.2008.08.003

Goodfellow, I., McDaniel, P., & Papernot, N. (2018). Making machine learning robust against adversarial inputs. Communications of the ACM, 61(7), 56–66. https://doi.org/10.1145/3134599

Grosse, K., Papernot, N., Manoharan, P., Backes, M., & McDaniel, P. (2017). Adversarial examples for malware detection. In Computer Security–ESORICS 2017: 22nd European Symposium on Research in Computer Security (pp. 62–79). Springer. https://doi.org/10.1007/978-3-319-66399-9_4

Hou, C. K. J., & Behdinan, K. (2022). Dimensionality reduction in surrogate modeling: A review of combined methods. Data Science and Engineering, 7(4), 402–427. https://doi.org/10.1007/s41019-022-00193-5

Jang-Jaccard, J., & Nepal, S. (2014). A survey of emerging threats in cybersecurity. Journal of Computer and System Sciences, 80(5), 973–993. https://doi.org/10.1016/j.jcss.2014.02.005

Jeeva, S. C., & Rajsingh, E. B. (2016). Intelligent phishing url detection using association rule mining. Human-Centric Computing and Information Sciences, 6(1), 10. https://doi.org/10.1186/s13673-016-0064-3

Jiang, P., Xiao, J., Li, D., Yu, H., Bai, Y., Guo, Y., & Chen, X. (2023). Detecting malicious websites from the perspective of system provenance analysis. IEEE Transactions on Dependable and Secure Computing, 21(3), 1406–1423. https://doi.org/10.1109/TDSC.2023.3277613

Karbab, E. B., Debbabi, M., Derhab, A., & Mouheb, D. (2018). MalDozer: Automatic framework for android malware detection using deep learning. Digital Investigation, 24, S48–S59. https://doi.org/10.1016/j.diin.2018.01.007

Lamina, O. A., Ayuba, W. A., Adebiyi, O. E., Michael, G. E., Samuel, O.-O. D., & Samuel, K. O. (2024). AI-powered phishing detection and prevention. Path of Science, 10(12), 4001–4010. https://doi.org/10.22178/pos.112-7

Lee, S.-W., Sidqi, H. M., Mohammadi, M., Rashidi, S., Rahmani, A. M., Masdari, M., & Hosseinzadeh, M. (2021). Towards secure intrusion detection systems using deep learning techniques: Comprehensive analysis and review. Journal of Network and Computer Applications, 187, 103111. https://doi.org/10.1016/j.jnca.2021.103111

Lezzi, M., Del Vecchio, V., & Lazoi, M. (2024). Using blockchain technology for sustainability and secure data management in the energy industry: Implications and future research directions. Sustainability, 16(18), 7949. https://doi.org/10.3390/su16187949

Ma, X., Wu, J., Xue, S., Yang, J., Zhou, C., Sheng, Q. Z., Xiong, H., & Akoglu, L. (2021). A comprehensive survey on graph anomaly detection with deep learning. IEEE Transactions on Knowledge and Data Engineering, 35(12), 12012–12038. https://doi.org/10.1109/TKDE.2021.3118815

Nair, M. M., Deshmukh, A., & Tyagi, A. K. (2024). Artificial intelligence for cybersecurity: Current trends and future challenges. In Automated Security and Computer Systems—Generative Systems (pp. 83–114). Wiley. https://doi.org/10.1002/9781394213948.ch5

Nazir, A., He, J., Zhu, N., Qureshi, S. S., Qureshi, S. U., Ullah, F., Wajahat, A., & Pathan, M. S. (2024). A deep learning-based novel hybrid CNN-LSTM architecture for efficiently detecting threats in the IoT ecosystem. Ain Shams Engineering Journal, 15(7), 102777. https://doi.org/10.1016/j.asej.2024.102777

Pascanu, R., Stokes, J. W., Sanossian, H., Marinescu, M., & Thomas, A. (2015). Malware classification with recurrent networks. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1916–1920). IEEE. https://doi.org/10.1109/ICASSP.2015.7178304

Pei, X., Yu, L., & Tian, S. (2020). AMalNet: A deep learning framework based on graph convolutional networks for malware detection. Computers & Security, 93, 101792. https://doi.org/10.1016/j.cose.2020.101792

Sharif, M., Bhagavatula, S., Bauer, L., & Reiter, M. K. (2016). Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 1528–1540). ACM. https://doi.org/10.1145/2976749.2978392

Sun, N., Ding, M., Jiang, J., Xu, W., Mo, X., Tai, Y., & Zhang, J. (2023). Cyber threat intelligence mining for proactive cybersecurity defense: A survey and new perspectives. IEEE Communications Surveys & Tutorials, 25(3), 1748–1774. https://doi.org/10.1109/COMST.2023.3273282

Trinh, N. B., Phan, T. D., & Pham, V.-H. (2022). Leveraging deep learning image classifiers for visual similarity-based phishing website detection. In Proceedings of the 11th International Symposium on Information and Communication Technology (SoICT 2022) (pp. 134–141). ACM. https://doi.org/10.1145/3568562.3568629

Yang, X., & Yan, J. (2022). On the arbitrary-oriented object detection: Classification-based approaches revisited. International Journal of Computer Vision, 130(5), 1340–1365. https://doi.org/10.1007/s11263-022-01618-4

Yuan, X., Li, C., & Li, X. (2017). DeepDefense: Identifying DDoS attack via deep learning. In Proceedings of the IEEE International Conference on Smart Computing (SMARTCOMP) (pp. 1–8). IEEE. https://doi.org/10.1109/SMARTCOMP.2017.7946998

Zhou, C., & Paffenroth, R. C. (2017). Anomaly detection with robust deep autoencoders. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 665–674). ACM. https://doi.org/10.1145/3097983.3098052