Comparative Performance Analysis of Neural Network–Based Techniques for Botnet Detection in Smart Grid Cyber-Physical Systems

Saraso David Lami[1], Agu Onyebueke Edward[2], Emmanuel Siman[3], Madugu Jeremiah Omanga[4]

Department of Computer Science, Kwararafa University Wukari[1],

Department of Computer Science, Federal University Wukari[2],

Department of Computer Science, Kwararafa University Wukari[3],

Department of Computer Science, Taraba State University Jalingo[4]

DOI: https://doi.org/10.51584/IJRIAS.2026.110200142

Received: 16 February 2026; Accepted: 21 February 2026; Published: 20 March 2026

ABSTRACT

The increasing integration of information and communication technologies into smart power grids has significantly improved operational efficiency but has also introduced critical cybersecurity vulnerabilities. Among emerging threats, botnet attacks pose a serious risk to smart grid cyber-physical systems due to their distributed, adaptive, and coordinated nature. This study presents a comparative performance analysis of three neural network–based techniques, Artificial Neural Networks (ANN), Deep Neural Networks (DNN), and Long Short-Term Memory (LSTM) networks, for detecting botnet-induced anomalies in smart grid environments. To guarantee fairness and repeatability, the models were trained and assessed using a publicly accessible smart-grid cyberattack dataset under similar preprocessing and experimental settings. Standard assessment measures, such as accuracy, precision, recall, F1-score, convergence behavior, and confusion matrix analysis, were used to gauge performance. The findings show that DNN produces better accuracy and overall classification stability through richer feature representations, whereas ANN offers a robust baseline with good recall. Because it can incorporate temporal relationships in smart-grid data, the LSTM model performs superior to both ANN and DNN on all measures, exhibiting greater accuracy and balanced detection capabilities. These results show that temporally aware deep learning models provide notable benefits for detecting coordinated botnet attacks and emphasize the significance of model architecture choices in cyber-physical intrusion detection. The study provides valuable insights for researchers and practitioners seeking effective data-driven security solutions for protecting critical smart grid infrastructure.

Keywords: Artificial Neural Networks (ANN), Deep Neural Networks (DNN), and Long Short-Term Memory (LSTM) networks, Smart-grid, Network Anomaly

INTRODUCTION

The modernization of electrical power systems through the integration of advanced information and communication technologies has given rise to the concept of the smart grid, a cyber-physical system that tightly couples digital control infrastructures with physical power generation, transmission, and distribution components (Al-Shetwi *et al*, 2025). While this transformation has significantly enhanced operational efficiency, reliability, and real-time situational awareness, it has simultaneously expanded the attack surface of power systems, exposing them to sophisticated cyber threats (Atıcı, & Tuna, 2025). Among these threats, botnet attacks represent one of the most severe and persistent challenges to smart-grid security due to their distributed nature, adaptability, and ability to exploit both cyber and physical vulnerabilities (Manias *et al*., 2024). Botnets can be orchestrated to launch coordinated attacks such as distributed denial-of-service (DDoS), false data injection, and command-and-control manipulation, potentially leading to cascading failures, grid instability, and large-scale power outages.

Traditional cybersecurity mechanisms, including signature-based intrusion detection systems and rule-driven firewalls, are increasingly inadequate in addressing the dynamic and evolving nature of botnet attacks in smart grid cyber-physical systems (Ren *et al*., 2025). These approaches typically rely on predefined attack patterns and static thresholds, making them ineffective against zero-day exploits, encrypted traffic, and adaptive attack strategies. Furthermore, the real-time operational constraints of smart grids demand detection mechanisms that not only achieve high accuracy but also minimize false negatives, as undetected intrusions may have catastrophic physical and economic consequences (Sahani *et al*., 2023). Consequently, there is a growing need for intelligent, data-driven security frameworks capable of learning complex patterns, adapting to evolving threats, and operating reliably within cyber-physical environments.

Deep learning and artificial intelligence have shown promise as ways to improve smart-grid cybersecurity in recent years (Bhuiyan, 2025). Because neural network-based models can extract high-dimensional feature representations, learn nonlinear relationships, and generalize across a variety of operating situations, they are especially well-suited for this field (Ullah *et al*., 2025). Due to their reliable performance and very cheap computational cost, Artificial Neural Networks (ANNs) have become popular baseline classifiers for intrusion detection (Suresh Babu, 2024). However, their shallow architectures often limit their capacity to capture intricate attack signatures embedded in complex cyber-physical interactions. Deep Neural Networks (DNNs), by contrast, extend this capability through hierarchical feature extraction, enabling more effective modeling of complex and abstract attack patterns (Hnamte *et al*., 2024). Nevertheless, both ANN and DNN architectures typically treat observations as independent samples, which restricts their ability to exploit temporal dependencies inherent in smart-grid operational data.

By explicitly modeling sequential and temporal links between data streams, Long Short-Term Memory (LSTM) networks, a specific type of recurrent neural networks, address this problem (Krichen & Mihoub, 2025). Because malicious activity sometimes appears as coordinated behaviors over time rather than individual abnormalities, this feature is very helpful for botnet identification. Long-range dependencies and changing attack dynamics that are challenging to identify with feedforward architectures alone can be captured by LSTM networks by preserving internal memory states and addressing the vanishing gradient issue. Despite the demonstrated effectiveness of these individual neural network models, existing studies often evaluate them in isolation, making it challenging to determine their relative strengths, limitations, and suitability for deployment in real-world smart-grid environments.

While ANN, DNN, and LSTM models have all been used separately for intrusion detection in smart grids and related fields, a systematic and cohesive comparison examination of their performance under consistent experimental settings is frequently lacking in the literature. It is challenging to get firm conclusions on the relative advantages and disadvantages of these models due to variations in datasets, assessment criteria, and preprocessing methods between research. Analyzing convergence behavior, confusion matrix features, and misclassification trends, all crucial for comprehending model dependability in safety-critical applications like smart grids, has also received little attention. Therefore, this study presents a comparative performance analysis of ANN, DNN, and LSTM models for botnet detection in cyber-physical smart grid systems.

RELATED LITERATURE

In order to identify cyber-physical intrusions in the SCADA system, Diaba *et al*. (2022) implemented an intrusion detection system that included the Gated Recurrent Unit and Convolutional Neural Network. To verify the effectiveness of the suggested intrusion detection model in a smart metering setting, numerous tests were carried out using a benchmark dataset. Current deep learning models are compared with parameters including accuracy, precision, and false-positive rate. Compared to current methods, the suggested concatenated approach achieves a detection accuracy of 98.84%. Using accuracy, precision, recall, and F1-score as assessment measures, Abdullahi *et al*. (2024) examined the efficacy of long-short-term memory (LSTM) and extreme gradient boosting (XGBoost) models for identifying cyberattacks inside CPS settings. They pointed out that a gas pipeline industrial control system dataset and benchmark datasets with a variety of attack types, such NetML-2020 and IoT-23, were used to evaluate these models. In a number of measures, the researchers found that both XGBoost and LSTM performed better than conventional methods like artificial neural networks (ANN) and

support vector machines (SVM). Lastly, they said that recommendations for future research paths in cyberattack detection for CPSs are included in the study's conclusion. In their study, Patel *et al*. (2024) used a Recurrent Neural Network (RNN) model to create an efficient intrusion detection and prevention system that targets botnet attacks in blockchain technology. They confirmed that this model could more accurately distinguish between malicious and normal network behaviors in decentralized environments. They clarified that, in contrast to conventional signature-based techniques, the RNN-based methodology was able to capture temporal patterns present in botnet traffic, enhancing detection rates. Additionally, the authors stated that their system was built to function in real-time, enabling timely mitigation of risks found before serious harm was done. Furthermore, they noted that incorporating deep learning into blockchain security frameworks could help address evolving attack strategies that conventional systems often miss. Finally, they suggested that their results demonstrated both feasibility and promise for broader application of machine learning techniques in enhancing distributed ledger security. Maiti and Dey (2024) stated to have created a deep reinforcement learning (DRL)-based defensive agent that can counteract these attacks by initiating suitable protection sequences. They also claimed to have formally verified the DRL agent's safety using reachability analysis to guarantee dependable performance. They contended that their method creates a new set of protection rules that effectively stymie current cyber-physical threats to grid operations and that their framework can be implemented in real time on GPU computers, allowing for quick execution and confirmation of protective actions. In order to improve smart grid resilience against fraudulent manipulation of sensor data and load demands, the authors stressed the practical value of their study. Hussain *et al.* (2024) used a Bi-Directional Long Short-Term Memory (Bi-LSTM) network to investigate anomaly detection in cyber-physical electric vehicle charging stations. They explained that this network could capture temporal patterns both forward and backward in time to better identify irregular behaviors in charging station data. They claimed that in order to improve detection performance, their data-driven anomaly detection (DDAD) model extracted important statistical features from EV charging system parameters and used feature selection. To validate the method, they produced a labeled dataset of both typical operations and simulated cyberattack scenarios. The authors stated that their Bi-LSTM-based approach achieved noticeably high accuracy when compared to baseline methods like traditional LSTM, multi-layer perceptron, support vector machine, and linear regression models, highlighting its efficacy for protecting EV charging infrastructure against various faults and attacks. In order to show the practicality of their concept, they also reported extensive testing in a hardware-in-the-loop intelligent cyber-physical system setting. Overall, the researchers argued that their approach strengthens the cyber-physical cybersecurity of distributed EV charging systems by providing a solid deep-learning framework. According to Dayarathne *et al*. (2025), the integration of decentralized renewable energy sources makes smart cyber-physical power systems more vulnerable to cybersecurity threats like replay attacks, denial of service attacks, and false data injection. The authors noted that in order to mitigate these risks, they created a hybrid security architecture that combines traditional cybersecurity techniques with deep learning models like CNN and LSTM to accurately identify anomalies in real-time grid data. They stated that their strategy included using cutting-edge pre-processing methods to improve feature extraction and simulating cyberattack scenarios on PSCAD datasets. The authors claimed that the integrated method achieved over 98 % detection accuracy, demonstrating its potential to bolster the resilience and stability of smart grids. They also highlighted implementation challenges, such as noisy data handling and the need for scalable, efficient models for real-world deployment. Sagar and Chandrasekaran (2025) investigated the application of deep learning models to enhance cyber-security in cyber-physical systems, concentrating on the detection and mitigation of different attacks utilizing complex algorithms. They explained that these models were used not just for CPS security but also to solve prediction problems in precision agriculture by forecasting rainfall from sensor data to assist surveillance systems. The authors noted that current systems faced significant challenges due to shifting technical conditions and inadequate cooperation in IoT solutions. They highlighted how their proposed approach improved prediction accuracy and resilience in smart grids and other CPS scenarios. Finally, Sagar and Chandrasekaran argued that integrating deep learning could substantially bolster CPS security frameworks and operational efficiency. In order to improve robust threat identification while maintaining data privacy across dispersed grid nodes, Xie *et al*. (2025) introduced a unique federated deep learning architecture that integrates a temporal convolutional network with multi-feature integration. In order to increase training efficiency and parameter security, the study described how to incorporate a gradient compression method utilizing an LSTM-based variational autoencoder. The authors emphasized that experimental validation demonstrates the better efficacy of their approach in identifying various cyberthreats when compared to current methods. According to Abou-Elasaad *et al.* (2025), their proposed AI-driven intrusion detection framework integrates recurrent neural

networks with support vector classifiers to effectively distinguish between normal and malicious traffic. The authors claimed that this hybrid model enhances real-time detection accuracy and robustness compared to traditional methods; they described preprocessing, temporal pattern analysis, and refined attack categorization as the core methodological stages; their evaluation on benchmark datasets showed near-perfect detection performance, which they interpreted as evidence of the system's potential to protect critical infrastructure; and they concluded that the suggested approach provides a scalable and resilient solution for securing smart grid communications. By transforming network traffic into spectrogram images and using Convolutional Neural Networks (CNNs) with explainable AI to enhance interpretability, Imtiaz *et al.* (2025) introduced an explainable deep learning model called XIoT to detect various Internet of Things (IoT) intrusion attacks over high-speed optical networks. They clarified that by providing insights into its decision-making process, XIoT was created to circumvent the real-time scalability and transparency constraints of conventional intrusion detection systems. According to the study, XIoT outperformed earlier models by achieving extremely high detection accuracies in testing spanning benchmark datasets including KDD CUP99, UNSW-NB15, and Bot-IoT. The authors indicated that the model's integration of explainability and optical network-optimized processing could enhance robust cybersecurity for large-scale IoT deployments. They concluded that XIoT offers a promising approach to scalable, interpretable intrusion detection suited to modern IoT environments.

In summary, the examined literature shows how neural network-based methods for intrusion and botnet detection in smart grid cyber-physical systems are becoming more and more successful. Studies utilizing ANN, DNN, and LSTM models separately have shown encouraging results. Nevertheless, the majority of current research assesses these models separately, frequently using various datasets, preprocessing techniques, and performance indicators, making it difficult to reach firm conclusions about their relative advantages and disadvantages. Comparative analyses that look at convergence behavior, misclassification features, and balanced performance across accuracy, precision, and recall under uniform experimental settings have also received little attention. These gaps highlight the need for a systematic and fair comparative evaluation of ANN, DNN, and LSTM architectures within a consistent framework. Addressing this need forms the central motivation of the present study, which aims to provide clearer insights into the suitability of these neural network models for effective botnet detection in smart grid environments.

RESEARCH FRAMEWORK

This study adopts a structured three-phase research framework designed to enable a rigorous comparative analysis of neural network–based techniques for botnet detection in smart grid cyber-physical systems. In the first phase, a high-quality smart-grid cyberattack dataset obtained from the Kaggle Machine Learning Repository is subjected to comprehensive preprocessing, including data cleaning, outlier handling, normalization, standardization, and feature extraction, to ensure consistency, reliability, and suitability for deep learning. The refined dataset is then randomly partitioned into training and validation subsets using a 70:30 split ratio, preserving the original statistical characteristics and class distribution to prevent sampling bias and support unbiased generalization assessment. Three neural network architectures, ANN, DNN, and LSTM networks, are implemented and trained in the second phase. These networks were chosen for their unique capacities to represent nonlinear connections or temporal dependencies seen in smart-grid data. To enable equitable and efficient learning across all approaches, model training is complemented by meticulous architectural design and hyperparameter optimization. In order to objectively measure detection efficacy and comparative strengths, the trained models are systematically assessed in the final phase using conventional performance metrics, such as accuracy, precision, recall, F1-score, and Receiver Operating Characteristic (ROC) analysis. Collectively, this framework ensures methodological rigor, reproducibility, and analytical clarity, providing a robust foundation for evaluating the relative suitability of ANN, DNN, and LSTM models for botnet detection in smart grid cyber-physical environments. The Framework for this study is presented in Figure 1.

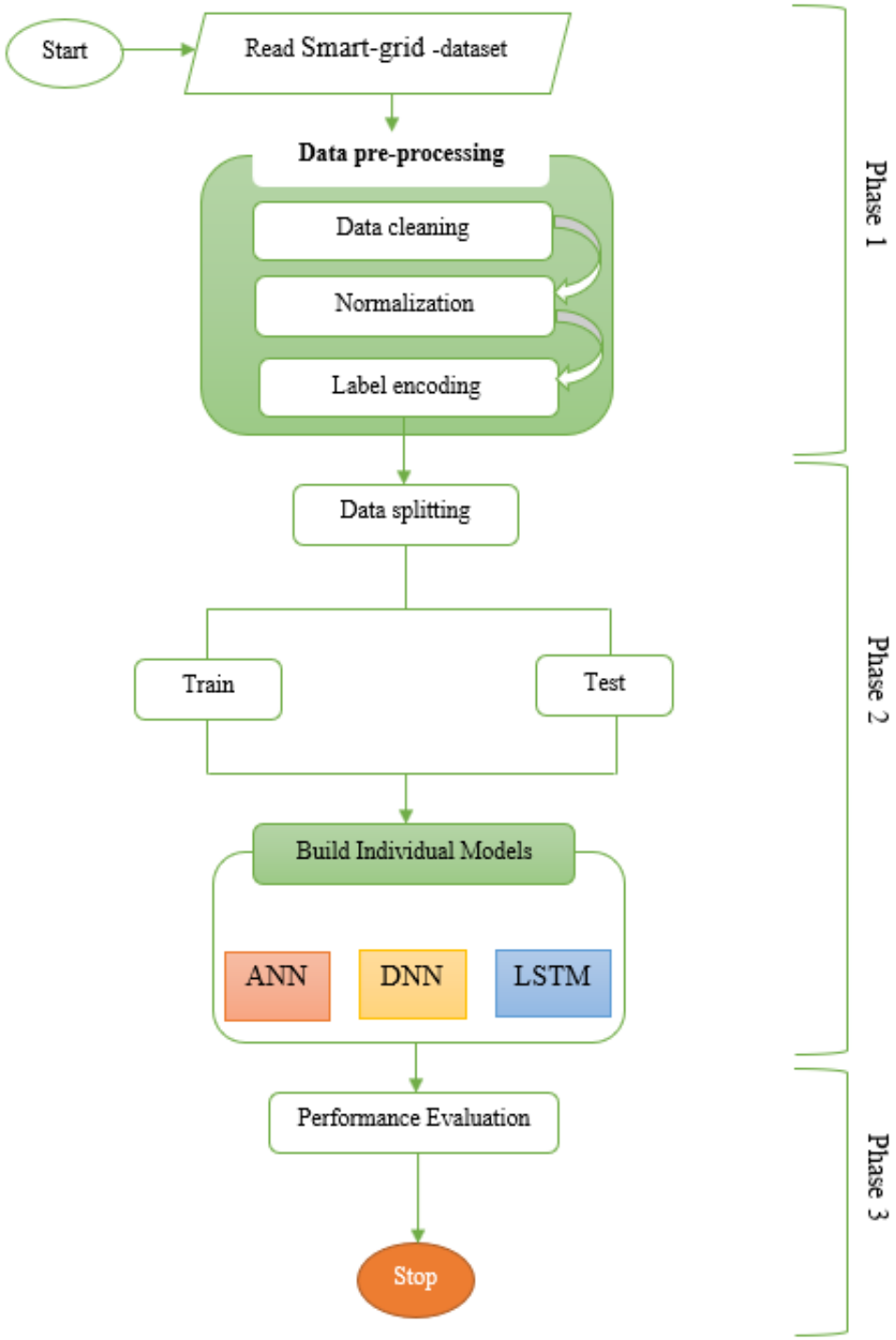

Figure 1: Research Framework

Deep Learning (DL) Algorithms in Botnet Attack

Deep learning is a subset of machine learning that employs multi-layered neural networks to automatically learn hierarchical feature representations from large and complex datasets. Its effectiveness is driven by advances in algorithms, increased computational power, and the availability of high-dimensional data, enabling the detection of intricate patterns and anomalies that traditional methods often miss. In smart power grids, deep learning is particularly valuable for identifying evolving and sophisticated botnet attacks that adapt to evade conventional security mechanisms. By analyzing large volumes of network traffic, sensor measurements, and operational data, deep learning models provide real-time threat detection and enhanced situational awareness. Consequently, deep learning offers a scalable and robust approach for strengthening cyber-physical security in smart grid environments.

Deep Learning Model Selection and Implementation

This study implements three deep learning models, selected for their complementary strengths in botnet detection. The LSTM model is employed for its effectiveness in modeling sequential and time-dependent network traffic patterns, while the ANN captures complex nonlinear relationships in smart-grid communication data. The DNN further enhances learning by extracting hierarchical and high-level feature representations. All models are implemented using Python-based deep learning frameworks such as TensorFlow and PyTorch. Their performance is optimized through a systematic hyperparameter tuning process to ensure fair and effective comparison.

Artificial Neural Networks (ANNs)

The strong ability of Artificial Neural Networks (ANNs) to learn patterns and identify anomalies in complicated smart-grid network data makes them an excellent fit for this investigation. ANNs are able to discriminate between typical and botnet-driven activity by examining traffic parameters including packet size, protocol type, timestamps, and source-destination information. They can respond to changing botnet tactics and generalize from previous attack behaviors thanks to their adaptive learning capability, which is essential in dynamic cyber-physical environments. The input, hidden, and output layers of the suggested ANN architecture allow for the derivation of intricate attack signatures thanks to nonlinear activation functions like ReLU. The ANN is a dependable baseline model for botnet identification in smart power grids since the output layer offers binary or multi-class categorization. The forward propagation procedure is the first step in the mathematical formulation of the ANN model. Each neuron in the hidden layer adds non-linearity by applying an activation function after computing the weighted sum of its inputs. The following is a possible mathematical expression for this:

Mathematically,

$$z^{(l)} = w^{(l)} x^{(l-l)} b^{(l)} \quad (1)$$

where $z^{(l)}$ is the weighted sum of inputs in layer l $w^{(l)}$ represents the weight matrix, $x^{(l-1)}$ is the input from the previous layer, and $b^{(l)}$ is the bias term. The activation function is applied to introduce non-linearity:

$$a^{(l)} = f(z^{(l)} \quad (2)$$

For this study, the ReLU activation function is chosen due to its efficiency in deep learning models:

$$fz=\max(0,z) \quad (3)$$

Once the network propagates the input through all layers, the final layer applies a softmax function if multi-class classification is needed:

$$\hat{y}_i = \frac{e^{zi}}{\sum_{j=l}^{n} e^{zj}} \quad (4)$$

where $\hat{y}_i$is the predicted probability for class i and n possible attack categories.

At the backpropagation and training stage, the model will be trained using supervised learning on the dataset. The loss function, typically cross-entropy for classification will be computed as follows:

$$J(W,b) = -\frac{l}{m}\sum_{i=l}^{m}\sum_{j=l}^{n} y_{i,j}\log(\hat{y}_{i,j}) \quad (5)$$

Where $y_{i,j}$ is the actual label, and $\hat{y}_{i,j}$ is the predicted probability. The backpropagation computes the gradient of the loss function for the weights and biases thus:

$$\partial J \partial W(l)\ \delta(l) a(l-1) T \quad (6)$$

$$\partial J \partial bl = \delta l \quad (7)$$

Where $\delta^{(l)}$ represents the error term computed at each layer using:

$$J\delta(l) = Wl + 1\delta l + 1^{\circ} f'(z(l)) \quad (8)$$

Weights are updated using gradient descent:

$$W^{(l)} \coloneqq W^{(l)} - a\frac{\partial J}{\partial W^{(l)}} \quad (9)$$

Where a is the learning rate.

Deep Neural Network (DNN)

Multiple hidden layers between the input and output layers allow for hierarchical feature learning in a Deep Neural Network (DNN), a sophisticated type of artificial neural network. High-level representations that capture intricate patterns are gradually transformed from raw input data by hidden layers using weighted connections and activation functions. This layered depth greatly improves the model's ability to identify complex links in big datasets. A DNN is particularly suited for assessing the dynamic and diverse data produced in smart-grid situations because of its capacity to recognize hidden properties and adjust to changing trends.

Mathematically

An input layer, several hidden layers, and an output layer are the mathematical components of a DNN. Each neuron in that layer receives weighted inputs from neurons in the layer above, applies an activation function, and transmits the result to the layer below.

Given the function $X = [x_1, x_2, x_3, x_4, \ldots x_n]$ the forward propagation in a DNN operates thus:

Linear Transformation: Each layer performs a weighted sum of inputs and adds a bias term thus:

$$Z^{(l)} = W^{(l)} A^{(l-l)} + b^{(l)} \quad (10)$$

Where $Z^{(l)}$ represents the pre-activation values at layer l, $W^{(l)}$ is the weight matrix while $A^{(l-1)}$ is the activation from the previous layer, and $b^{(l)}$ is the bias term.

Activation Function: To introduce non-linearity, an activation function $f(\cdot)$ is applied to DNN algorithm.

$$A^{(l)} = f(^{(l)}) \quad (11)$$

The introduction of the activation function helps mitigate the vanishing gradient problem.

Loss Function: A loss function is used to compare the prediction vector $\hat{Y}$ produced by the output layer with the ground truth Y. Cross-entropy loss is used for binary categorization (attack vs. regular traffic):

$$L(Y, \hat{Y}) = -\sum_{i=l}^{m} [Y_{i,} \log(\hat{Y}_i) + (1 - Y_i) \log(1 - \hat{Y}_i)] \quad (12)$$

Where m is the number of training.

Backpropagation and Optimization: Using backpropagation and gradient descent, the model parameters W and b are updated. The chain rule is used to calculate the gradients of the loss function about each weight:

$$\frac{\partial L}{\partial W^{(l)}} = \frac{\partial L}{\partial A^{(l)}} . \frac{\partial A^{(l)}}{\partial Z^{(l)}} . \frac{\partial Z^{(l)}}{\partial W^{(l)}} \quad (13)$$

The weight update will be performed using the Adam optimization algorithm.

$W(l)=W(l)-\eta\partial L\partial W(l)$ (14)

Where η is the learning rate.

Prediction and Decision Making: At the final layer, binary classification is made using the sigmoid activation function.

$$\hat{Y} = 1 - \frac{1}{1 + e^{-Z^{(L)}}} \quad (15)$$

At this point, a threshold is applied to determine whether the input corresponds to an attack or not.

Long-Short Term Memory (LSTM) Algorithm

An improved version of a recurrent neural network called the Long Short-Term Memory (LSTM) algorithm was created to describe sequential input while maintaining long-term dependencies. The vanishing gradient issue that traditional RNNs frequently have is successfully resolved by LSTM through the use of memory cells and gating techniques. This feature makes the model ideal for identifying temporal trends in smart grid data as it allows it to preserve and apply historical data across long periods. As a result, LSTM offers improved performance in identifying abnormalities that change over time, including coordinated botnet activity. The behavior of an LSTM cell is described by the following equations:

$$c_{in}^{t} = \tanh(W_{xc}x^{t} + W_{hc}h^{t-1} + b_c) \quad (16)$$

$$i^{t} = \text{sigmoid}(W_{xi}x^{t} + W_{hi}h^{t-1} + b_i) \quad (17)$$

$$o^{t} = \text{sigmoid}\left(W_{xo}x^{t} + W_{ho}h^{t-1} + b_o + b_{forget}\right) \quad (18)$$

$$f^{t} = \text{sigmoid}\left(W_{xf}x^{t} + W_{hf}h^{t-1} + b_f\right) \quad (19)$$

$$c^{t} = f^{t}c^{t-1} + i^{t}c_{in}^{t} \quad (20)$$

$$h^{t} = o^{t}\tanh(c^{t}) \quad (21)$$

Where $W \in \mathbb{R}^{ls \times ls}, x^t, h^t, o^t, f^t, c^t, b \in \mathbb{R}^{ls}$. ls is a hyperparameter, called the LSTM size, and is defined upfront by design as constant among all cells.

Model Performance Evaluation

Evaluating the security model's performance is crucial for this investigation. The assessment metrics used in this work to determine how successfully the model identifies botnet attacks in the smart power grid are accuracy, precision, recall, and F1-score.

i. **Accuracy**: This determines the proportion of correctly classified cases (attack and normal occurrences) in relation to all instances, is a crucial measure of the model's overall correctness. Its mathematical definition is as follows:

$$\boldsymbol{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (22)$$

TP (True Positives) denotes the number of botnet attacks that were correctly identified, TN (True Negatives) denotes the quantity of correctly identified normal instances, FP (False Positives) denotes normal events that were incorrectly classified as attacks, and FN (False Negatives) denotes botnet attacks that were incorrectly classified as normal events.

ii. **Precision:** Out of all cases that were expected to be attacked, the precision measure determines the proportion of botnet attacks that are successfully identified. This statistic is extremely important in security systems since false positives, or incorrect alarms, can lead to unnecessary resource allocation. The precision formula is as follows:

$$\boldsymbol{Precision} = \frac{TP}{TP + FP} \quad (23)$$

The method successfully reduces false alarms, guaranteeing that threats that are identified are, in fact, botnet assaults, when the precision parameter is high.

iii. **Recall**: In order to evaluate the system's ability to detect genuine botnet attacks, recall, also known as sensitivity or true positive rate, measures the proportion of successfully identified assaults among all actual attack cases. The recall formula is expressed as follows:

$$\boldsymbol{Recall} = \frac{TP}{TP + FN} \quad (24)$$

The chance of security breaches brought on by unreported threats is decreased when the model has a high recall value, which indicates that it successfully identifies the majority of botnet assaults.

iv. **F1-score:** The F1-score provides a fair assessment of the model by considering both accuracy and recall. By calculating the harmonic mean of accuracy and recall, it ensures that both false positives and false negatives are taken into account. The F1-score is defined as follows:

$$\mathbf{F1}_{\boldsymbol{-Score}} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (25)$$

One of the most important metrics in botnet attack mitigation is the F1-score, which shows how well the system balances detecting botnet attacks with reducing false alarms.

The total of these metrics provides a comprehensive evaluation of the system's effectiveness in detecting botnet attacks in this study. Accuracy offers an overall performance indicator, recall focuses on recognizing actual threats, precision ensures a decrease in false positives, and the F1-score ensures a trade-off between these two variables.

IMPLEMENTATION AND RESULT DISCUSSION

The implementation details and experimental outcomes of the neural network-based botnet detection models applied to the smart grid dataset are presented in this part. The training procedure, performance assessment, and comparative study of ANN, DNN, and LSTM designs are covered. The efficacy, dependability, and appropriateness of each model for smart grid cyber-physical security are evaluated by a critical analysis of the findings.

Experimental Setup

The experimental setup for this study was designed to support the development, training, and evaluation of deep learning models for detecting botnet-induced anomalies in a smart power grid environment. A smart-grid cyberattack dataset sourced from the Kaggle Machine Learning Repository was used, containing records of both normal grid behavior and attack-related instability indicators. These data were employed to construct classification models capable of distinguishing legitimate grid operations from botnet-driven anomalies. All

experiments were conducted on a 64-bit Windows system powered by an Intel® Core™ i5-3630QM processor operating at 2.40 GHz with 4 GB RAM. Despite the moderate hardware configuration, the platform was sufficient for executing preprocessing, training, and evaluation tasks. Model implementation was carried out using Python 3.8 within the Anaconda distribution, selected for its robust support for data science workflows and seamless integration with machine learning libraries.

Dataset Description and Visualization

60,000 records with 14 features, 13 continuous numerical variables that represent torque parameters, power measurements, generator indicators, and a stability index, as well as a categorical target label that separates normal from botnet-induced unstable states make up the smart-grid cyber-physical dataset used in this study. Initial integrity tests verified that there were no missing values, confirming that the dataset was appropriate for direct analytical processing. A degree of imbalance between normal and anomalous traffic was found by analyzing the class label distribution, which has consequences for model evaluation and training. Significant correlations between a number of process variables were revealed using exploratory data visualizations, suggesting hidden structural links within the smart-grid system. The variance structure of the dataset was further explained by Principal Component Analysis, which demonstrated that a small number of components accounted for a significant amount of the overall variability. Together, these descriptive statistics and visual analyses gave a thorough grasp of the dataset's features and influenced further modeling and preprocessing techniques for efficient botnet identification. Figure 2 presents the dataset visualization of this study.

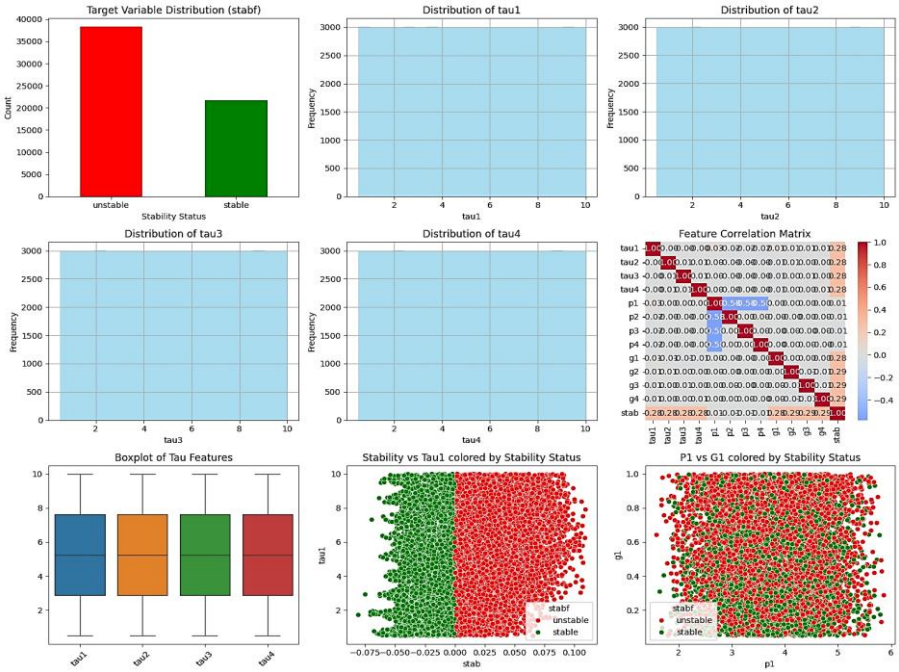

Figure 2: Visualization of Cleansed Dataset

Parameter Tuning

The parameter tuning procedures applied to each algorithm evaluated in the study was conducted systematically to identify the optimal hyperparameters for ANN, DNN, and LSTM models. Through iterative experimentation, the most effective configurations for accurate smart-grid botnet detection were determined.

ANN Parameter Tuning

The ANN parameter tuning procedure is shown in Table 1, which emphasizes how various hyperparameter configurations affect learning behavior and classification performance in the cyber-physical smart-grid context. To find the ideal configuration, several combinations of learning rates, activation functions, hidden-layer sizes, and optimization techniques were methodically assessed. While preserving generalization and avoiding overfitting, the chosen settings produced the least amount of training loss.

Table 1: LSTM Parameter Tuning

Parameter	Final Setting	Justification
Input Size	14 neurons	14 dataset features
Hidden Layers	2	Consistent with shallow ANN baseline
Neurons (Layer 1)	64	Enough capacity without overfitting
Neurons (Layer 2)	32	Gradual compression
Activation	ReLU	As specified in paper
Output Layer	1 neuron	Binary classification
Output Activation	Sigmoid	Binary classification
Loss Function	Binary Cross-Entropy	As described
Optimizer	Adam	Stable and efficient
Learning Rate	0.001	Standard stable LR for ANN
Batch Size	32	Works well on small hardware
Epochs	50–70	ANN converges faster
Dropout	0.3 (after each hidden layer)	Prevent overfitting
L2 Regularization	0.0001	Improve generalization
Weight Initialization	He Normal	For ReLU

DNN Parameter Tunning

The DNN's parameter adjustment is shown in Table 2, where more hidden layers increase the model's complexity and allow for the extraction of more abstract representations from the smart-grid stability indicators. To minimize vanishing gradients and lower generalization error, the tuning procedure methodically investigated various depth levels, neuron counts per layer, dropout rates, batch sizes, and learning rates.

Table 2: DNN Parameter Tuning

Parameter	Final Setting	Justification
Input Size	14 neurons	Feature count
Hidden Layers	4	True “deep” representation
Neurons (Layer 1)	128	Rich feature extraction
Neurons (Layer 2)	64	Dimensional reduction
Neurons (Layer 3)	32	Hierarchical compression
Neurons (Layer 4)	16	**ABSTRACT** representation
Activation	ReLU	Mitigates vanishing gradients
Output Layer	1 neuron	Binary classification
Output Activation	Sigmoid	As stated
Loss Function	Binary Cross-Entropy	As in paper

Parameter	Final Setting	Justification
Optimizer	Adam	Mentioned in equations
Learning Rate	0.0005	Lower LR for deeper net stability
Batch Size	64	Stabilizes deeper training
Epochs	80–100	Needs more training than ANN
Dropout	0.4 (after Layer 1 & 2), 0.3 (after Layer 3)	Reduce overfitting
L2 Regularization	0.0005	Improve generalization
Batch Normalization	Yes (after each hidden layer)	Stabilizes oscillations

LSTM Parameter Tuning

The Long Short-Term Memory (LSTM) network, which was used to take advantage of temporal dependencies and sequential patterns pertinent to botnet activity in smart-grid operations, shows the parameter tuning results in Table 3. In order to ascertain how these factors affected the model's capacity to maintain long-range dependencies without succumbing to gradient degradation, the tuning procedure evaluated changes in memory cell dimension, number of LSTM layers, recurrent dropout, sequence length, and learning rate.

Table 3: LSTM Parameter Tuning

Parameter	Final Setting	Justification
Input Shape	(10, 14)	10-step temporal window
LSTM Layers	2 stacked LSTM layers	Capture long-range dependencies
LSTM Units (Layer 1)	128	High memory capacity
LSTM Units (Layer 2)	64	Gradual abstraction
Recurrent Activation	Sigmoid	Standard LSTM
Activation	Tanh	Standard cell activation
Recurrent Dropout	0.3	Prevent temporal overfitting
Dropout	0.4	Generalization
Dense Layer	32 neurons (ReLU)	Feature refinement
Output Layer	1 neuron	Binary classification
Output Activation	Sigmoid	Binary
Loss Function	Binary Cross-Entropy	Consistent
Optimizer	Adam	Stable
Learning Rate	0.0003	Lower for RNN stability
Batch Size	64	Stabilizes gradient updates
Epochs	100–120	Slower convergence
Gradient Clipping	1.0	Prevent exploding gradients

Comparison of final hyperparameter tuning for the three models is presented in Table 4.

Table 4: Comparison of final hyperparameter tuning

Model	LR	Batch Size	Epochs	Layers	Dropout	Best Strength
ANN	0.001	32	60	2 hidden	0.3	Fast baseline
DNN	0.0005	64	90	4 hidden	0.4	Nonlinear depth
LSTM	0.0003	64	110	2 LSTM + Dense	0.4	Temporal modeling

Models Convergence

Figure 3's convergence results demonstrate different but consistent learning behaviors for each of the three models. Despite its shallow design, the ANN shows effective learning and strong generalization as it converges quickly with a smooth and monotonic loss decrease. Because of its greater representational capacity and capacity to catch intricate nonlinear patterns while preserving training stability, the DNN shows a quicker initial loss reduction with moderate oscillations. The LSTM, on the other hand, shows constant alignment between training and validation loss but converges more slowly because of its sequential modeling and gating methods. All models, with ANN delivering effective baseline learning, DNN offering improved nonlinear discrimination, and LSTM excelling in capturing temporal dependencies pertinent to botnet detection in smart-grid cyber-physical systems, achieve steady convergence without overfitting overall.

The model's convergence during training for the three models is illustrated in Figure 3 thus:

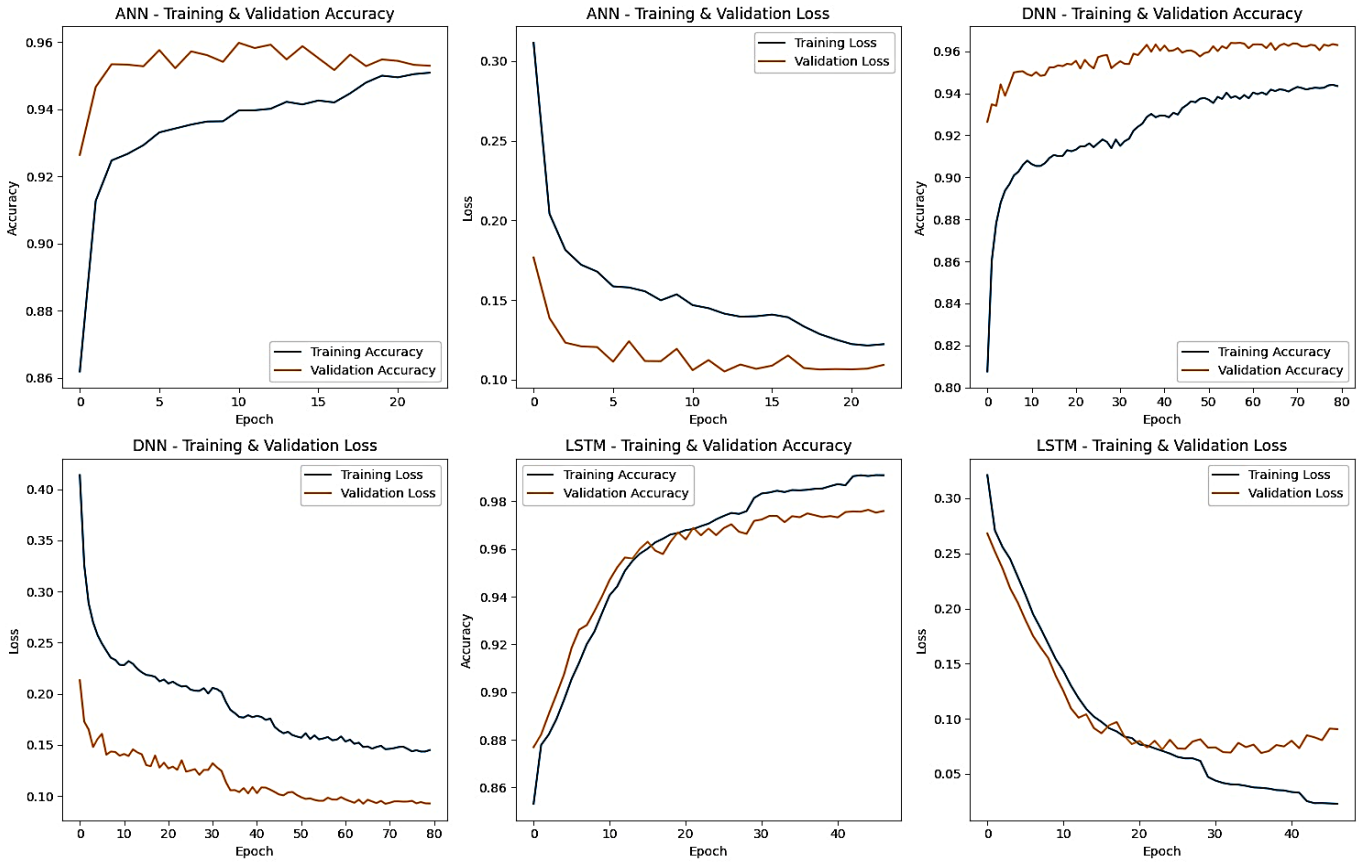

Figure 3: Model Convergence During Training

Confusion Matrix

The confusion matrices for the ANN, DNN, and LSTM models provide a detailed view of each model's classification behaviour across normal and botnet-affected operating states. The confusion matrix for the three models is illustrated in Figure 4 to 6.

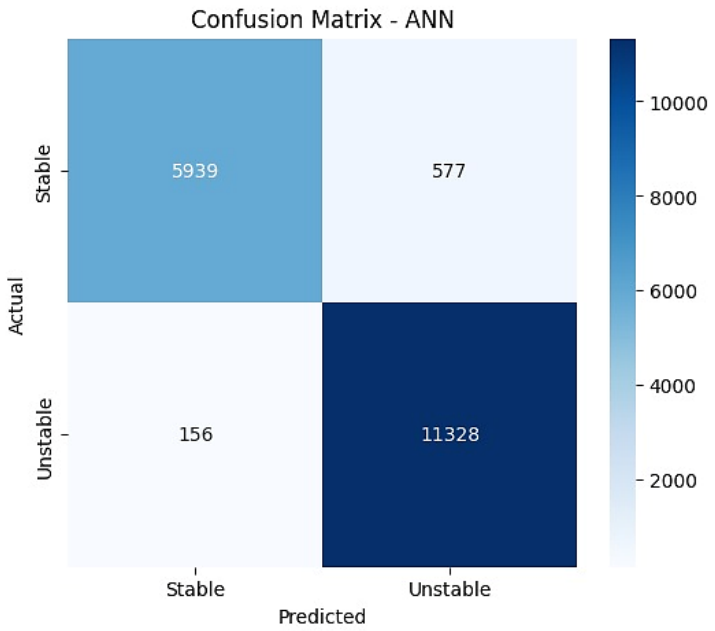

Figure 4: ANN Confusion Matrix

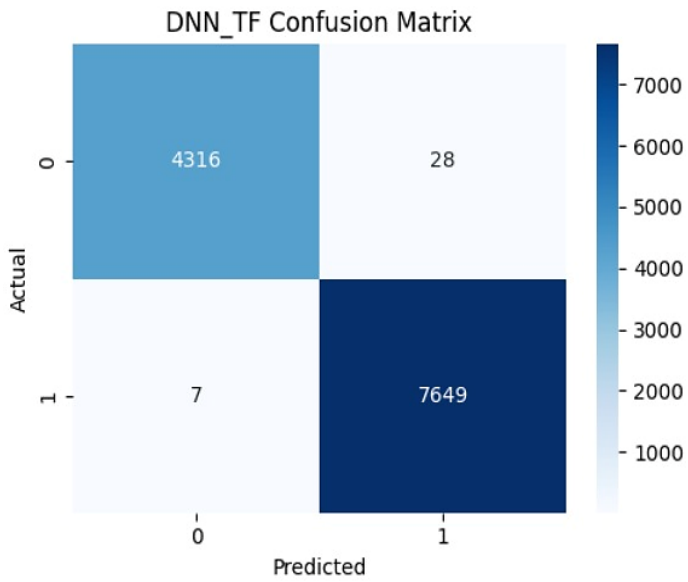

Figure 5: DNN Confusion Matrix

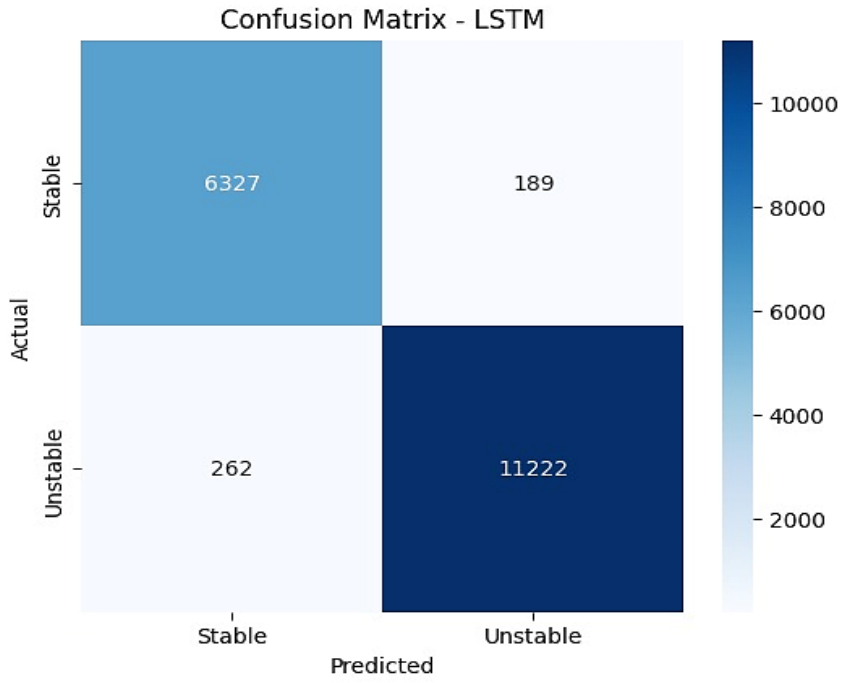

Figure 6: LSTM Confusion Matrix

Figure 4 distinguishes between *Stable* and *Unstable* smart-grid operating states, where Stable denotes normal, secure grid conditions without cyber interference, and Unstable represents compromised states typically caused by botnet-driven attacks that disrupt system dynamics. The non-attack class is represented by Stable in the confusion matrix, whereas the assault class is represented by Unstable. Correct classification denotes effective intrusion detection. Strong diagonal dominance is shown in all assessed models, indicating excellent accuracy in distinguishing between normal and abnormal circumstances. Although the ANN offers dependable baseline performance, it occasionally struggles to identify nuanced attack fingerprints, as seen by its somewhat higher false-negative rate. By lowering both false positives and false negatives, the DNN enhances detection and shows its improved capacity to represent intricate nonlinear connections. The LSTM's improved capacity to capture temporal patterns linked to coordinated or developing botnet assaults is demonstrated by the fact that it obtains the most balanced confusion matrix with few misclassifications.

Evaluation Metrics Result Validation

Table 5: Model's Performance Results

Models	Accuracy	Precision	Recall	F1-Score
ANN	0.9593	0.9515	0.9864	0.9687
DNN	0.9638	0.9604	0.9838	0.9720
LSTM	0.9749	0.9834	0.9772	0.9803

The efficacy of the three neural network models in identifying botnet activity within the smart grid cyber-physical system is confirmed by the assessment metrics shown in Table 5. With an accuracy of 95.93% and a high recall of 98.64%, the ANN performs well overall, demonstrating its efficacy in properly recognizing the bulk of assault events, but with somewhat lesser precision. By achieving better accuracy (96.38%) and precision (96.04%), the DNN outperforms this baseline, demonstrating its improved capacity to represent intricate nonlinear connections and lower false positives. With the best accuracy (97.49%), precision (98.34%), recall (97.72%), and F1-score (98.03%), the LSTM surpasses both ANN and DNN in every parameter, exhibiting a better balance between detection sensitivity and classification reliability. The elevated F1-score of the LSTM confirms its robustness in handling class imbalance while maintaining consistent performance. Collectively, these results substantiate the comparative advantage of deeper and temporally aware architectures for botnet detection in smart grid environments.

Model Performance Comparison

A strong advancement in detecting capabilities from ANN to DNN and finally to LSTM can be seen in the performance comparison of the three neural network models shown in Table 5. Although its comparatively lower precision predicts a larger risk of false positives, the ANN obtains a solid baseline performance with an accuracy of 95.93% and high recall (98.64%), suggesting successful identification of botnet operations. Because of its improved capacity to grasp intricate nonlinear feature interactions, the DNN enhances overall classification performance, achieving greater accuracy (96.38%) and precision (96.04%) while keeping a comparable recall. With the greatest accuracy (97.49%), precision (98.34%), recall (97.72%), and F1-score (98.03%), the LSTM model performs better than both ANN and DNN on all assessment criteria. This superior performance highlights the LSTM's effectiveness in learning temporal dependencies inherent in smart-grid data, leading to more balanced and reliable detection. Overall, the results confirm that increasing architectural depth and temporal modeling capability significantly enhances botnet detection performance in smart grid cyber-physical systems.

CONCLUSION AND FUTURE DIRECTION

A thorough comparison of neural network-based methods for botnet attack detection in smart grid cyber-physical systems is presented in this study, specifically on ANN, DNN, and LSTM models. To guarantee fairness and repeatability, the models were assessed using a carefully selected smart-grid cyberattack dataset under uniform preprocessing and training settings. The experimental findings showed that although all three models performed well in terms of detection, their efficacy differed depending on the architectural complexity and learning capacity. The ANN demonstrated a robust baseline with good recall, demonstrating its capacity to identify the majority of assault cases, despite relatively low accuracy. By capturing more intricate nonlinear feature interactions, the DNN increased accuracy and F1-score, improving overall classification balance. The LSTM model consistently outperformed both ANN and DNN across all evaluation metrics, highlighting the importance of temporal dependency modeling in identifying coordinated and evolving botnet behaviors within smart-grid environments.

The results demonstrate that robust deep learning architectures, especially those that can identify sequential patterns present in cyber-physical data streams, greatly enhance smart grid cybersecurity. The LSTM model's higher performance highlights its applicability in situations where attack behaviors appear gradually rather than as isolated incidents. Furthermore, the thorough examination of convergence behavior, confusion matrices, and

performance metrics offers insightful information on model dependability and misclassification tendencies, which are crucial factors for safety-critical infrastructures like power grids. Overall, by providing a transparent, data-driven comparison of neural network models in a smart-grid setting, our study adds to the expanding body of knowledge on intelligent intrusion detection.

Even with these advancements, there are still a number of avenues for further study. In order to improve model generalizability, future research may:

i. Investigate the incorporation of more temporal and contextual characteristics as well as the usage of larger and more varied datasets.
ii. Examining complex architectures like transformer networks or attention-based recurrent models may increase detection robustness and accuracy.

REFERENCES

1. Abdullahi, M., Alhussian, H., Aziz, N., Abdulkadir, S. J., Alwadain, A., Muazu, A. A., & Bala, A. (2024). Comparison and investigation of AI-based approaches for cyberattack detection in cyber-physical systems. *IEEE Access, 12*, 31988–32004. https://doi.org/10.1109/ACCESS.2024.3370436
2. Abou-Elasaad, M. M., Sayed, S. G., & El-Dakroury, M. M. (2025). Smart grid intrusion detection system based on AI techniques. Journal of Cybersecurity & Information Management, 15(2), 195. https://doi.org/10.54216/JCIM.150215
3. Al-Shetwi, Ali Q., et al. (2025) "Latest advancements in smart grid technologies and their transformative role in shaping the power systems of tomorrow: An overview." *Progress in Energy* 7.1: 012004.
4. Atıcı, S., & Tuna, G. (2025). Impact of cybersecurity attacks on electrical system operation. In *Cyber Security Solutions for Protecting and Building the Future Smart Grid* (pp. 117-160). Elsevier.
5. Bhuiyan, T. (2025). AI in Smart Grid Cybersecurity: A Systematic Review of Machine Learning and Deep Learning Approaches against False Data Injection and Other Emerging Attacks. *Journal of Computer Science and Technology Studies*, *7*(8), 1207-1295.
6. Dayarathne, M. A. S. P., Jayathilaka, M. S. M., Bandara, R. M. V. A., Logeeshan, V., Kumarawadu, S., & Wanigasekara, C. (2025). Mitigating cyber risks in smart cyber-physical power systems through deep learning and hybrid security models. *IEEE Access, 13*, 37474–37492. https://doi.org/10.1109/ACCESS.2025.3545637
7. Diaba, S. Y., Shafie-khah, M., & Elmusrati, M. (2022). On the performance metrics for cyber-physical attack detection in smart grid. *Soft Computing, 26*(23), 13109–13118. https://doi.org/10.1007/s00500-022-06761-1
8. Hnamte, V., Najar, A. A., Nhung-Nguyen, H., Hussain, J., & Sugali, M. N. (2024). DDoS attack detection and mitigation using deep neural network in SDN environment. *Computers & Security*, *138*, 103661.
9. Hussain, A., Yadav, A., & Ravikumar, G. (2024). Anomaly detection using bi-directional long short-term memory networks for cyber-physical electric vehicle charging stations. IEEE Transactions on Industrial Cyber-Physical Systems. https://doi.org/10s.1109/TICPS.2024.3437349
10. Imtiaz, N., Wahid, A., Ul Abideen, S. Z., Kamal, M. M., Sehito, N., Khan, S., Virdee, B. S., Kouhalvandi, L., & Alibakhshikenari, M. (2025). A deep learning-based approach for the detection of various Internet of Things intrusion attacks through optical networks. Photonics, 12(1), 35. https://doi.org/10.3390/photonics12010035
11. Krichen, M., & Mihoub, A. (2025). Long short-term memory networks: A comprehensive survey. *AI*, *6*(9), 215.
12. Maiti, S., & Dey, S. (2024). *Smart grid security*: A verified deep reinforcement learning framework to counter cyber-physical attacks. arXiv. https://doi.org/10.48550/arXiv.2409.15757
13. Manias, D. M., Saber, A. M., Radaideh, M. I., Gaber, A. T., Maniatakos, M., Zeineldin, H., ... & El-Saadany, E. F. (2024). Trends in smart grid cyber-physical security: Components, threats and solutions. *IEEE Access*.
14. Patel, N. K., Anagha, N., & BJ, S. K. (2024). Effective intrusion detection and prevention system of botnet attack in blockchain technology using recurrent neural network. In *Proceedings of the IEEE Conference* (pp. 1–6). IEEE. https://doi.org/10.1109/ciscon62171.2024.10696133

15. Ren, S., Chen, S., & Zhang, Q. (2025). Autonomous Threat Detection and Response through Actor–Critic Reinforcement Learning.
16. Sagar, M., & Chandrasekaran, V. (2025). An exploration of utilising deep learning models in the realm of cyber-physical systems. *International Journal of Services, Economics and Management, 16*(4/5), 578–606. https://doi.org/10.1504/IJSEM.2025.148484
17. Sahani, N., Zhu, R., Cho, J. H., & Liu, C. C. (2023). Machine learning-based intrusion detection for smart grid computing: A survey. *ACM Transactions on Cyber-Physical Systems*, *7*(2), 1-31.
18. Suresh Babu, G. N. (2024). A Novel ANN-Based Support Vector Machine for improving Classification Accuracy in Intrusion Detection Systems. *Journal of Computational Analysis & Applications*, *33*(2).
19. Ullah, F., Ullah, I., Khan, K., Khan, S., & Amin, F. (2025). Advances in deep neural network-based hyperspectral image classification and feature learning with limited samples: a survey. *Applied Intelligence*, *55*(6), 370.
20. Xie, R., Wang, B., & Xu, X. (2025). A novel federated deep learning for intrusion detection in smart grid cyber-physical systems. *Engineering Applications of Artificial Intelligence, 162*(Part B), Article 112404. https://doi.org/10.1016/j.engappai.2025.112404