ICTP 2025 – 6th IEEE International Conference on Telecommunications and Photonics
18–19 December 2025, Dhaka, Bangladesh

Advancing Military Cybersecurity: A Scalable Ensemble Network Intrusion Detection System Framework with SHAP Analysis for Military Operations

Md. Tofael Ahmed Bhuiyan
Southeast University, Dhaka, Bangladesh
tofael1104@gmail.com

Md. Abdur Rahman
Southeast University, Dhaka, Bangladesh
2021200000025@seu.edu.bd

Arif Hossen
Research and Development Department, MetaHeed
arif@metaheed.com

Md Ruhul Rabbi
Risk Management Department, Green CyberSec Ltd.
ruhul.rabbi@greencybersec.net

Abdul Kadar Muhammad Masum*
Southeast University, Dhaka, Bangladesh
akmmasum@yahoo.com

***Abstract*- This study improves cybersecurity measures for military networks through a more advanced Network Intrusion Detection System (NIDS) using machine learning and deep learning approaches. The hybrid ensemble consists of XGBoost, Random Forest, CatBoost, and BiLSTM models, all trained on the Kaggle NSL-KDD dataset. Recursive Feature Elimination is employed for feature selections, while Grid Search is employed to optimize hyperparameters. With an accuracy of 99.78 percent, this system has a low rate of false alarms and demonstrates effective processing efficiency to operate in real time. It also includes comprehensive Exploratory Data Analysis (EDA) and improved model explainability with SHAP-based explainable artificial intelligence. All evaluations support the model's scalability to generalize across attack types. Overall, the results of this study represent a meaningful contribution to the NIDS literature and moves the impact of machine learning in improving cybersecurity in critical infrastructures forward through ensemble modeling, optimized learning, and explainable AI.**

Keywords- Network Intrusion Detection System, Machine Learning, Deep Learning, XGBoost, SHAP, Feature Selection, Cybersecurity, Real-Time Detection.

I. INTRODUCTION

With the rapid growth of digital infrastructure, effective intrusion detection systems (IDS) are critical to detect attacks against networks. The traditional approaches of IDS, which include signature-based and rule-based systems, struggle to identify novel and sophisticated attacks successfully [1]. In comparison, machine learning (ML) and deep learning (DL) approaches identify patterns in the network traffic, thus enabling better accuracy and versatility in detecting intrusions [2]. In particular, gradient-boosting methods, such as XGBoost, and ensemble methods, such as Random Forest (RF), have shown great success in classifying traffic by distinguishing malicious traffic from benign traffic [3, 4]. In addition to this, hybrid models, such as CNN-LSTM approaches, increase detection capabilities by leveraging temporal and spatial correlations in the network data [5, 6].

Random Forest is an ensemble learning method that constructs multiple decision trees and aggregates their outputs through majority voting for classification or averaging for regression [7, 8]. When applied to the NSL-KDD dataset, a benchmark for intrusion detection, RF achieved strong accuracy and generalization, proving effective for military network analysis [9]. XGBoost, in contrast, employs gradient boosting to optimize model parameters, achieving high accuracy, recall, and F1-score while minimizing false alarms, which is vital for reliable intrusion detection systems [10]. Its performance further improves with Recursive Feature Elimination [11].

In hybrid deep learning models CNN-LSTM, convolutional neural networks learned spatial features, while long short-term memory (LSTM) networks learned sequential dependencies [12,13]. These models learn features from both packet-level data and temporal traffic patterns, outperforming traditional intrusion detection systems (IDSs) in detecting complex attack types [14]. Support Vector Machines (SVMs) perform well for binary classification of network traffic with low computational cost using suitable feature selection methods [15,16].

Connecting to previous knowledge of failure and fraud detection in electrical and financial systems [17,18], this research presents an IDS built from multiple algorithms including RF, XGBoost, CatBoost, Gradient Boosting, SVM, and a Voting Classifier. The 41 features of traffic dynamics were mined from TCP/IP dump data to emulate a LAN environment using the NSL-KDD dataset [19].

The work makes the following key contributions:

- Multi-Algorithm IDS Framework: An intrusion detection system is developed leveraging state-of-the-art algorithms (XGBoost, RF, CatBoost, etc.), achieving high detection performance across diverse attack scenarios.
- Feature Selection Optimization: Through the use of RF, SVM, and XGBoost in conjunction with Recursive Feature Elimination (RFE), the improve accuracy while lowering computational complexity.
- Hyperparameter Tuning: Grid Search improves generalization and resource efficiency by optimizing model parameters.
- Explainability with SHAP: SHapley Additive exPlanations (SHAP) is integrated to interpret model decisions, thereby enhancing transparency and trust in the intrusion detection system (IDS).
- Defense-Grade Adaptation: The suggested ensemble structure has been set up and verified in situations that mimic high-volume, mission-critical defense communications.
- Scalable Ensemble Integration: Accuracy, interpretability, and scalability are all fairly matched when XGBoost, Random Forest, CatBoost, and BiLSTM are combined.
- Novel Explainability Integration: This study stresses explainable cybersecurity by using SHAP analysis to highlight decision transparency and practical insights for intelligence and defense operations, in addition to attaining higher accuracy.

The remainder of the document is organized as follows: Section II summarizes related work in network intrusion detection; Section III describes the proposed approach, including model architecture and datasets; Section IV presents experimental results and performance analysis; and Section V concludes the document with recommendations for future research.

II. Literature Review

Benchmark datasets have advanced machine learning and deep learning for Network Intrusion Detection Systems, yet scalability, generalizability, and real-time performance remain constrained by the methodological and practical limitations of existing techniques.

M. H. Bhuiyan et al. [13] proposed a Deep Neural Network (DNN) based NIDS to detect stealth and polymorphic attacks with 99% accuracy, besting hybrid architectures such as CNN+BiLSTM and GRU+RNN. However, high throughput or real-time applications can face issues due to high computational complexity and not having explicit optimization for features. Z. Ahmad et al. [14] indicated that RNNs and autoencoders can work well for detection, but lack of standardized assessment and benchmarking processes can lead to uncomparable or replicable results.

M. M. Hoque et al. [15] utilized a CNN+BiLSTM combination to achieve high accuracy in identifying Trojan horse traffic, though evaluation on broader attack types was lacking. Alzahrani et al. [16] developed an ML-based NIDS for Software-Defined Networks, reducing features from 41 to 5 while maintaining 95.95% accuracy, though excessive reduction may have excluded key discriminatory features. Monir et al. [17] explored IDS designs using a Click modular router, noting that jitter and transmission rates adversely indicated interference. Bhuiyan et al. [18] proposed an IoT-based home automation framework using multiple sensors for security but found limited resilience to IoT cyberattacks. Rahman et al. [19] implemented permission-based feature selection for detecting Android malware, where overlapping permissions between malicious and benign apps caused higher false positives.

To tackle these issues, an optimized multi-algorithm ensemble Network Intrusion Detection System is proposed using XGBoost, Random Forest, deep neural networks, and Bidirectional long-short term memory with Recursive Feature Elimination. The system achieves high accuracy, minimal false positives, and real-time efficiency. SHAP enhances explainability by identifying critical features, ensuring scalability and reliability in NIDS research.

III. Methodology

Data plays a fundamental role in machine learning as the basis for predictive algorithms, which include both regression and classification techniques, as illustrated in Fig. 1 below.

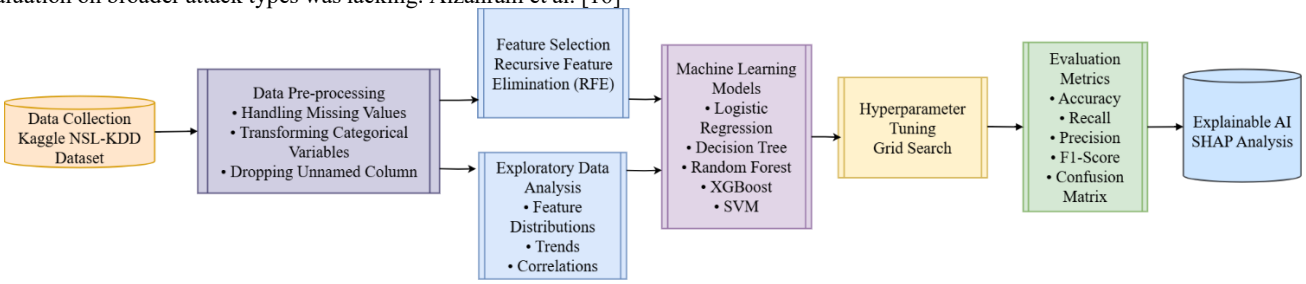


Fig. 1: Proposed Methodology

A. Dataset

The dataset from Kaggle [10] consists of 25,192 rows of network traffic statistics, with 39 columns capturing both regular and anomalous activity. The dataset is provided in CSV format and is well suited to training an intrusion detection model because of its diversity.

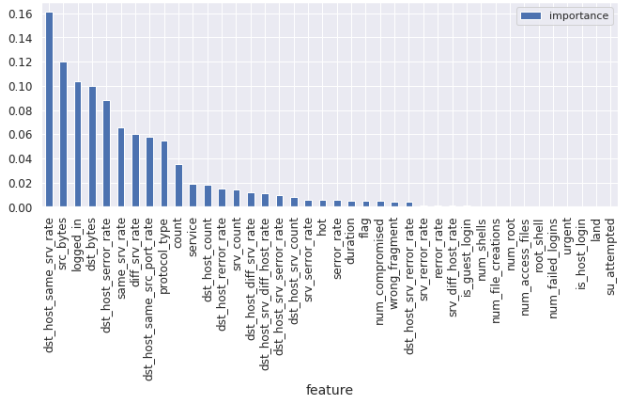


Fig.2: Features analysis

Fig. 2 illustrates the improved visual understanding of feature interactions by visualizing patterns and correlations in the data. These visualizations help expand our knowledge and aid in identifying the important variables needed in order to evaluate machine learning models on network intrusion detection systems (NIDS).

B. Data Pre-processing

In order to conduct reliable machine-learning analysis, data must be pre-processed. This section discusses important methods for preparing the dataset to perform intrusion detection. In this study, the following pre-processing methods were used:

- Handling Missing Values: Imputation and deletion were used to resolve missing values in the dataset, guaranteeing machine learning algorithms a smooth learning experience [11].
- Transforming Categorical Variables: One-hot encoding is used to prepare categorical variables for machine learning by converting them into binary vectors (1 for presence, 0 for absence) [12].
- Dropping Unnamed Column: To ensure the model focuses on relevant data, the unidentified column- which was probably unnecessary and automatically generated-was eliminated in order to streamline the dataset and lower noise.

C. Feature Selection

To identify the most pertinent subset, the model-based feature selection technique Recursive Feature Elimination (RFE) removes features systematically. The pairwise correlations between features selected by RFE are shown in Fig. 3 (“Selected Features Correlation Matrix”). The heatmap highlights duplicate characteristics through connections ranging from strong positive to negative. For instance, a strong correlation between src_bytes and dst_bytes suggests potential information overlap. This visualization enhances model interpretability and performance by ensuring the final feature set is predictive and minimally redundant, supporting RFE’s goal of optimized feature selection.

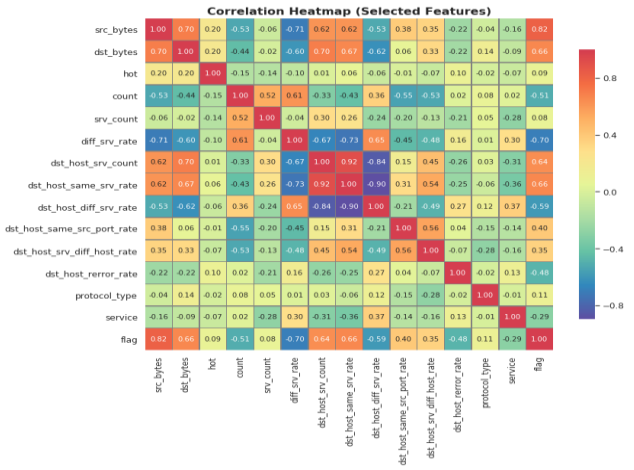


Fig.3: Selected Feature Correction Matrix

D. Machine Learning Models

Machine learning models improve cybersecurity analysis for Network Intrusion Detection Systems. Logistic Regression classifies normal versus abnormal traffic, Decision Trees enhance explainability, Random Forest boosts accuracy, XGBoost detects subtle intrusion patterns, and SVM separates complex patterns. Testing identifies the optimal detection model for best performance output.

E. Classification Metrics

To assess the performance of our NIDS using machine learning, we used the following essential metrics for evaluation: accuracy, recall, precision, F1-score, and the Confusion Matrix was used to view the details of the predictions.

1) Confusion Matrix: The Confusion matrix segments true negatives (TN), false positives (FP), false negatives (FN), and test positives (TP) to demonstrate prediction accuracy and identify kinds of mistakes. This study shows the types of categorization errors to improve the model.

2) Accuracy: The percentage of true positives as well as real negatives to all instances is known as accuracy, and it gauges how accurate our models are overall.

$$Accuracy = \frac{\{TP + TN\}}{\{TP + FP + TN + FN\}} \quad (1)$$

3) Precision: The precision of our models is determined by dividing the number of genuine positive forecasts by the total number of positive predictions.

$$Precision = \frac{\{TP\}}{\{TP + FP\}} \quad (2)$$

4) Recall (Sensitivity): The model's recall, also known as sensitivity, is determined by dividing the total number of real positive instances by the fraction of genuine positives.

$$Recall = \frac{\{TP\}}{\{TP + FN\}} \quad (3)$$

5) F1-Score: The F1-Score is a fair indicator of the model's ability to recognize beneficial as well as detrimental occurrences.

$$F1 = 2 * \frac{\{Precision \cdot Recall\}}{\{Precision + Recall\}} \quad (4)$$

IV. Result Analysis

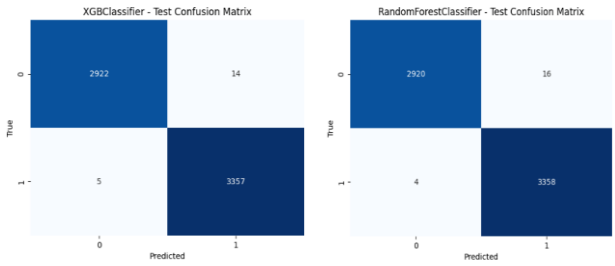
Based on a detailed study of accuracy in machine learning algorithms for NIDS, the best performing algorithm was XGBoost at 99.78% accuracy, followed by Random Forest at 99.71%, Decision Tree at 99.36%, and Logistic Regression at 95.55%. Conversely, BernoulliNB and AdaBoost demonstrated reasonable accuracy, whereas linear models such as Logistic Regression and LinearSVC had somewhat worse performance. According to this investigation, deep learning models like BiLSTM and ensemble techniques like XGBoost, Random Forest, LGBM, and CatBoost show outstanding precision, recall, and F1 scores in addition to high accuracy when it comes to identifying abnormalities and regular traffic. Simplified models and the Support Vector Classifier (SVC) trail a little, suggesting room for improvement. Precision, recall, and F1-score are anomaly-based metrics that were used to evaluate the proposed NIDS. Precision measures the number of correctly identified attacks among the predictions, while recall is about the actual attacks detected. The F1-score balances both. They collectively estimate detection reliability for rare critical intrusions within military network environments. These results provide important information for choosing the best algorithm to improve network intrusion detection systems' cybersecurity.

TABLE I. Performace Analysis of vaious ML

Classifier	Accuracy	Anomaly Precision	Anomaly Recall	Anomaly F1	Normal Precision	Normal Recall	Normal F1
XGB Classifier	0.9978	0.9958	0.9972	0.9973	0.9958	0.9972	0.9973
Random Forest Classifier	0.9971	0.9980	0.9959	0.9969	0.9964	0.9982	0.9973
LGBM Classifier	0.9970	0.9983	0.9952	0.9968	0.9958	0.9985	0.9972
CatBoost Classifier	0.9968	0.9983	0.9949	0.9966	0.9956	0.9985	0.9970
Voting Classifier	0.9960	0.9979	0.9935	0.9957	0.9944	0.9982	0.9963
Gradient Boosting Classifier	0.9946	0.9945	0.9939	0.9942	0.9946	0.9952	0.9949
Decision Tree Classifier	0.9936	0.9929	0.9935	0.9932	0.9943	0.9938	0.9940
KNeighbors Classifier	0.9933	0.9952	0.9905	0.9928	0.9917	0.9958	0.9938
SVC	0.9898	0.9878	0.9905	0.9891	0.9917	0.9893	0.9905
AdaBoost Classifier	0.9878	0.9881	0.9857	0.9869	0.9875	0.9896	0.9886
Logistic Regression	0.9555	0.9608	0.9431	0.9519	0.9511	0.9664	0.9587
Linear SVC	0.9555	0.9640	0.9397	0.9517	0.9485	0.9694	0.9588
RFE Logistic Regression	0.9320	0.9357	0.9172	0.9264	0.9289	0.9450	0.9369
BernoulliNB	0.8962	0.9416	0.8287	0.8815	0.8646	0.9551	0.9076
BiLSTM	0.9968	0.9986	0.9946	0.9966	0.9953	0.9988	0.9970
DNN	0.9754	0.9895	0.9002	0.9434	0.9500	0.9880	0.9687

TABLE II. HYPERPARAMETER DETAILS

Model	Key Hyperparameters	Values/Ranges	Optimization Method	Special Notes
XGBoost	learning_rate, max_depth, n_estimators	0.01-0.3, 3-10, 50-200	Bayesian Opt	GPU acceleration enabled
Random Forest	n_estimators, max_depth, min_samples_split	100-500, 10-50, 2-10	GridSearchCV	Gini impurity, bootstrap=True
LightGBM	num_leaves, learning_rate, feature_fraction	15-255, 0.01-0.1, 0.7-1.0	Random Search	categorical_feature handling
CatBoost	iterations, depth, l2_leaf_reg	500-2000, 4-10, 1-10	Automated	Built-in categorical processing
Voting Classifier	weights, voting	Optimized, ['hard','soft']	Custom	RF+XGB+LR combination
Gradient Boosting	loss, learning_rate, n_estimators	deviance/huber, 0.05-0.2, 50-200	GridSearch	subsample=0.8
Decision Tree	max_depth, min_samples_leaf	5-30, 1-10	RandomizedSearch	splitter='best'
KNN	n_neighbors, weights, p	3-15, ['uniform','distance'], [1,2]	GridSearch	metric='minkowski'
SVC	C, kernel, gamma	0.1-10, ['rbf','poly'], ['scale','auto']	Bayesian	probability=True
AdaBoost	n_estimators, learning_rate	50-200, 0.01-1.0	GridSearch	base_estimator=DT
Logistic Regression	penalty, C, solver	['l1','l2'], 0.001-10, ['liblinear','saga']	Hyperopt	class_weight='balanced'
Linear SVC	penalty, C, loss	['l1','l2'], 0.001-10, ['hinge','squared_hinge']	Random Search	dual=False
RFE Logistic	n_features_to_select, step	10-50, 1-5	RFECV	estimator=LogisticRegression
BernoulliNB	alpha, binarize	0.1-1.0, 0.0-1.0	GridSearch	fit_prior=True
BiLSTM	units, dropout, epochs	64-256, 0.1-0.5, 20-100	Keras Tuner	return_sequences=True
DNN	hidden_layers, dropout	2-5, 0.1-0.5	Hyperband	batch_norm=True



(a). XGBoost (b). Random Forest

Fig.4: Confusion Matrix of the best models

XGBoost and Random Forest perform well, whereas Support Vector Machine (SVM) performs worse, according to Fig. 4, which compares F1-score, accuracy, precision, and recall among models.

V. EXPLAINABLE AI

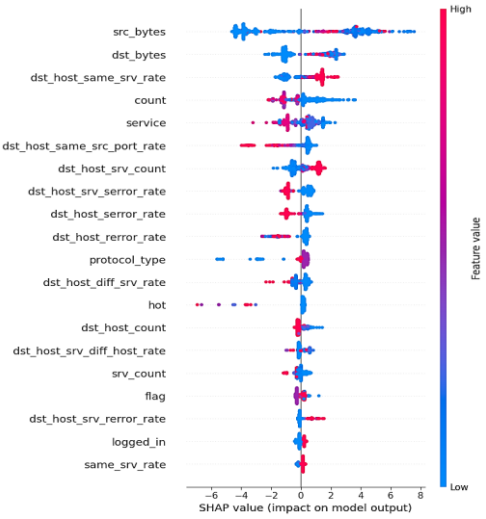


Fig.5: SHAP Summery plot of the XGBoost model

This SHAP summary plot shows how each feature affects the XGBoost model's predictions. Features are displayed on the y-axis, and the x-axis displays the SHAP values that represent each feature's contribution to the model's output. A single prediction is represented by each dot, which is colored according to the feature value (blue for low, red for high), illustrating how various values affect the forecast's direction. The large range of SHAP values indicates that features near the top, such src_bytes and dst_bytes, have the most overall impact. The plot helps users understand which inputs influence model choices and how they are made by giving them a clear, interpretable picture of feature significance and behavior.

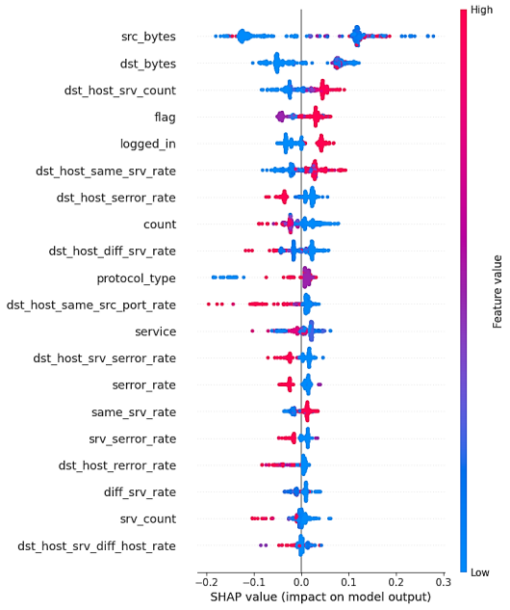


Fig.6: SHAP Summery plot of the Random Forest model

By listing each feature on the y-axis and showing its corresponding SHAP values on the x-axis, this SHAP summary

plot for a Random Forest model shows how different features affect the model's predictions. Positive values indicate a higher likelihood of the predicted class, while negative values indicate a lower one. A single prediction is represented by each dot, which is colored according to the feature value (red for high, blue for low), demonstrating the impact of varying feature magnitudes on results. Recognizable attributes with wide SHAP value distributions (for example, src_bytes, dst_bytes, and dst_host_same_srv_rate) show their considerable influence on model decisions. This graphic is beneficial for understanding how the Random Forest model operates since it provides a succinct, clear summary of feature importance and behavior.

TABLE III. COMPARISON TABLE WITH EXISTING WORK

Author	Dataset	Method	Accuracy	XAI
Bhuiyan et al. [13]	Not specified (benchmark datasets implied)	DNN	99%	×
Hoque et al. [15]	Real-world datasets	CNN+BiLSTM	High (not quantified)	×
Alzahrani [16]	NSL-KDD(41 features reduced to 5)	ML-integrated NIDS in SDN	95.95%	×
proposed methodology	**Kaggle Network Intrusion Detection**	**XGBoost (with RFE and SHAP)**	**99.78%**	**SHAP**

VI. CONCLUSION

To advance beyond current methodologies while achieving computational efficiency and lower false alarm rates for real-world operationalization, this study presents a robust ensemble framework based on XGBoost, Random Forest, CatBoost, and BiLSTM models, which incorporates optimized features. The proposed ensemble framework achieves maximum accuracy of 99.78% on the Kaggle dataset. A meaningful step toward protecting military and important network infrastructures has been taken with the addition of Recursive Feature Elimination (RFE) and Grid Search optimization to improve model performance, and SHAP-based explainability that reveals important feature contributions and fosters trust. Future work will focus on expanding the framework in terms of robustness and scalability to next generation NIDS, against dynamic threat environments, with adaptive learning techniques and real-time data streams, exploring lightweight models for edge computing, federated learning for privacy-preserving multi-domain deployments, and improving SHAP interpretations through counterfactual analysis.

REFERENCES

[1] "What is Network Intrusion Detection System (NIDS)? – Sapphire.net," *Sapphire.net*. [Online]. Available: https://www.sapphire.net/blogs-press-releases/nids/. [Accessed: Aug. 14, 2025].

[2] "Discover thousands of collaborative articles on 2500+ skills," *LinkedIn*. [Online]. Available: https://www.linkedin.com/pulse/random-forest-explained-regression-classification-tasks-sellahewa. [Accessed: Aug. 14, 2025].

[3] G. S. Fuhnwi, M. Revelle, and C. Izurieta, "Improving network intrusion detection performance: An empirical evaluation using extreme gradient boosting (xgboost) with recursive feature elimination," in *Proc. 2024 IEEE 3rd Int. Conf. AI in Cybersecurity (ICAIC)*, 2024, pp. 1–8.

[4] G. Mohiuddin, Z. Lin, J. Zheng, *et al.*, "Intrusion detection using hybridized meta-heuristic techniques with weighted xgboost classifier," *Expert Syst. Appl.*, vol. 232, p. 120596, 2023.

[5] M. Alhussein, K. Aurangzeb, and S. I. Haider, "Hybrid CNN–LSTM model for short-term individual household load forecasting," *IEEE Access*, vol. 8, pp. 180544–180557, 2020.

[6] S. K. Wanjau, G. M. Wambugu, A. M. Oirere, and G. M. Muketha, "Discriminative spatial-temporal feature learning for modeling network intrusion detection systems," *J. Comput. Secur.*, vol. 32, no. 1, pp. 1–30, 2024.

[7] J. Han and W. Pak, "Hierarchical LSTM-based network intrusion detection system using hybrid classification," *Appl. Sci.*, vol. 13, no. 5, p. 3089, 2023.

[8] F. Guo, H. Jiao, X. Zhang, Y. Zhou, and H. Feng, "Information security network intrusion detection system based on machine learning," in *Proc. 2024 Int. Conf. Data Sci. Netw. Secur. (ICDSNS)*, IEEE, 2024, pp. 1–4.

[9] M. Khodaskar, D. Medhane, R. Ingle, A. Buchade, and A. Khodaskar, "Feature-based intrusion detection system with support vector machine," in *Proc. 2022 IEEE Int. Conf. Blockchain Distributed Syst. Secur. (ICBDS)*, IEEE, 2022, pp. 1–7.

[10] M. Y. Turaba, M. Hasan, N. I. Khan, and H. A. Rahman, "Fraud detection during financial transactions using machine learning and deep learning techniques," in *Proc. 2022 Int. Conf. Commun., Comput., Cybersecurity, Informatics (CCCI)*, IEEE, 2022, pp. 1–8.

[11] S. Bin Akter, T. Sarkar Pias, S. Rahman Deeba, J. Hossain, and H. A. Rahman, "Ensemble learning-based transmission line fault classification using phasor measurement unit (PMU) data with explainable AI (XAI)," *PLOS One*, vol. 19, no. 2, e0295144, 2024.

[12] "NSL-KDD," *Kaggle.com*. [Online]. Available: https://www.kaggle.com/datasets/hassan06/nslkdd. [Accessed: Aug. 14, 2025].

[13] M. H. Bhuiyan, K. Alam, K. I. Shahin, and D. M. Farid, "A deep learning approach for network intrusion classification," in *Proc. 2024 IEEE Region 10 Symp. (TENSYMP)*, 2024, pp. 1–6, doi: 10.1109/TENSYMP61132.2024.10752251.

[14] Z. Ahmad, A. S. Khan, W. S. Cheah, J. bin Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Trans. Emerg. Telecommun. Technol.*, vol. 32, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:225153435.

[15] M. M. Hoque, K. Alam, M. F. Monir, and M. T. Habib, "Deep learning-based Trojan detection in network traffic: A CNN–BiLSTM approach," in *Proc. 2024 IEEE 100th Veh. Technol. Conf. (VTC2024-Fall)*, IEEE, 2024, pp. 1–5.

[16] A. O. Alzahrani and M. J. F. Alenazi, "Designing a network intrusion detection system based on machine learning for software defined networks," *Future Internet*, vol. 13, p. 111, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:235363621.

[17] M. F. Monir, R. Uddin, and D. Pan, "Implementation of a click-based IDS on SDN–NFV architecture and performance evaluation," in *Proc. 2021 IEEE Int. Black Sea Conf. Commun. Netw. (BlackSeaCom)*, 2021, pp. 1–6, doi: 10.1109/BlackSeaCom52164.2021.9527751.

[18] M. H. Bhuiyan, R. K. Ahad, A. J. Haque, M. F. Monir, and T. Ahmed, "An affordable and effective IoT-based home automation and security system for everyone," in *Proc. IEEE EUROCON 2023 - 20th Int. Conf. Smart Technol.*, 2023, pp. 325–330, doi: 10.1109/EUROCON56442.2023.10198937.

[19] R. Rahman, M. R. Islam, A. Ahmed, M. K. Hasan, and H. Mahmud, "A study of permission-based malware detection using machine learning," in *Proc. 2022 15th Int. Conf. Secur. Inf. Netw. (SIN)*, 2022, pp. 1–6, doi: 10.1109/SIN56466.2022.9970528.