Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 29 (2026), No. 8, pp. 3137–3147
DOI : 10.47974/JDMSC-2698

A discrete mathematical framework for network intrusion detection with applications to cryptographic network security

Ajay Kumar [†]
Amit Kumar Gupta [*]
Department of Computer Science and Engineering
Manipal University Jaipur
Jaipur 303007
Rajasthan
India

Priya Mathur [§]
Department of Mathematics
Poornima Institute of Engineering and Technology
Jaipur 302022
Rajasthan
India

Chaitanya Singh [‡]
Department of Computer Engineering
Vidhyadeep University
Surat 394110
Gujarat
India

Vipin Tiwari [@]
Department of Computer Science and Engineering
Symbiosis Institute of Technology (SIT)
Symbiosis International (Deemed University)
Pune 412115
Maharashtra
India

[†] *E-mail:* kumar.ajay@jaipur.manipal.edu
[*] *E-mail:* amit.gupta@jaipur.manipal.edu; (Corresponding Author)
dramitkumargupta1983@gmail.com
[§] *E-mail:* drpriyamathur21@gmail.com
[‡] *E-mail:* er.chaitanyasingh@gmail.com
[@] *E-mail:* vipintiwari1@gmail.com

Abstract

As computer networks expand, robust Intrusion Detection Systems (IDS) are crucial. Traditional methods struggle with zero-day attacks and lack interpretability. This study uses the NSL-KDD dataset and three machine learning models: Random Forest, attention-based Multilayer Perceptron (MLP), and hybrid CNN–BiLSTM. Models were evaluated on accuracy, error rate, ROC–AUC, and confusion matrix. The hybrid CNN–BiLSTM achieved 77.84% accuracy; attention-based MLP reached 0.9076 AUC. SHAP (Shapley Additive Explanations) analyzed key features in the Random Forest model, enhancing transparency. Results demonstrate the framework's improved real-time intrusion detection performance with increased explainability and practical network security applicability.

Subject Classification: *Primary 93A30, Secondary 49K15.*

Keywords: *Intrusion detection system, NSL-KDD dataset, CNN-BiLSTM, Attention MLP, Explainable AI, SHAP, Network security.*

1. Introduction

The growth of digital communication, cloud computing, and IoT has increased cyber threats like DoS, probing, and unauthorized access, emphasizing the need for intrusion detection systems (IDS) [1], [2]. IDS methods include signature-based, effective for known attacks but weak against zero-day threats [3], and anomaly-based, which detect novel attacks but have high false positives [4]. Machine learning techniques (Decision Trees, SVM, KNN, Random Forest) improve detection but require manual feature engineering and struggle with complex patterns [5]. Deep learning models like CNN and LSTM automatically extract features and capture spatial–temporal dependencies, enhancing performance [6], [7]. Hybrid CNN–BiLSTM architectures further improve IDS accuracy and learning capability [8], [9].

2. Related Work

Machine learning (ML) for intrusion detection evolved from early anomaly-based IDS frameworks by [1] and [2]. Traditional classifiers like Decision Trees, SVM, Bayesian models, and Random Forest showed reliable results but depended on manual feature engineering and lacked temporal modeling [5]. Deep learning architectures such as CNN and LSTM/RNN enabled automatic feature extraction and improved detection accuracy [6], [7]. Hybrid CNN–LSTM models enhanced spatial–temporal learning [8], but limited interpretability hinders adoption in security-critical settings [9]. Explainable AI methods like LIME [10] and SHAP [11] address this, with SHAP effective in cybersecurity [13], [14]. Using the NSL-KDD dataset [12], this study explores hybrid modeling, ROC analysis, and explainability to build a robust IDS framework [15], [16], emphasizing the need for mathematically structured models to capture complex attack patterns [21], [22].

3. Dataset Description

The original KDD Cup 1999 dataset was refined to remove redundancy and bias, resulting in the improved NSL-KDD benchmark [17]. It contains labeled traffic records with 41 features and predefined KDDTrain+ and KDDTest+ splits, ensuring fair evaluation and realistic detection of unseen attacks.

4. Proposed Methodology and Mathematical Formulation

This section presents an intrusion detection framework combining traditional ML and CNN–BiLSTM, evaluating Random Forest, attention-based MLP, and CNN–BiLSTM for accuracy, robustness, and transparency.

4.1 *Data Preprocessing*

The NSL-KDD dataset contains categorical (protocol_type, service, flag) and numerical features. Categorical attributes are label encoded, while numerical features are standardized using z-score normalization. Class labels are binarized (0: normal, 1: attack). The predefined KDDTrain+ and KDDTest+ splits are used for training and testing, preserving unseen attack types in the test set to simulate zero-day attacks.

4.2 *Problem Definition*

Let

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N} \tag{1}$$

denote the NSL-KDD dataset, where

$$\mathbf{x}_i \in \mathbb{R}^{41} \tag{2}$$

represents the feature vector of the i-th network connection, and

$$y_i \in \{0,1\} \tag{3}$$

denotes the corresponding class label, with 0 indicating normal traffic and 1 indicating malicious traffic. The objective of the intrusion detection system is to learn a mapping that effectively distinguishes normal and attack traffic, remains robust to unseen attacks, and ensures interpretability [18].

$$f: \mathbb{R}^{41} \rightarrow \{0,1\} \tag{4}$$

4.3 *Random Forest Classifier*

Random Forest uses bootstrap sampling and random features to enhance generalization [19]. Given a set of T decision trees $\{h_t(\mathbf{x})\}_{t=1}^{T}$, the final prediction is obtained through majority voting:

$$\hat{y} = \text{mode}\big(h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_T(\mathbf{x})\big) \tag{5}$$

Random Forest serves as a baseline and supports model explainability due to its transparent structure.

4.4 *Attention-Based Multilayer Perceptron (MLP)*

An attention-based MLP models nonlinear tabular traffic data, enabling fast training while capturing key feature interactions.

4.4.1 Attention Mechanism

Given an input feature vector $\mathbf{x} \in \mathbb{R}^{41}$, attention weights are computed as:

$$\boldsymbol{\alpha} = \text{softmax}(\mathbf{W}_a\mathbf{x} + \mathbf{b}_a) \tag{6}$$

where $\mathbf{W}_a$ and $\mathbf{b}_a$ are learnable parameters. The attention-weighted representation is obtained by:

$$\tilde{\mathbf{x}} = \boldsymbol{\alpha} \odot \mathbf{x} \tag{7}$$

allowing the model to emphasize security-relevant features while suppressing less informative ones.

4.4.2 MLP and Output Layer

The weighted features are passed through a multilayer perceptron:

$$\mathbf{h}_1 = \sigma(\mathbf{W}_1\tilde{\mathbf{x}} + \mathbf{b}_1), \mathbf{h}_2 = \sigma(\mathbf{W}_2\mathbf{h}_1 + \mathbf{b}_2) \tag{8}$$

where $\sigma(\cdot)$ denotes the ReLU activation function. The final output is computed using a sigmoid activation:

$$\hat{y} = \sigma(\mathbf{W}_o\mathbf{h}_2 + b_o) \tag{9}$$

where $\hat{y} \in [0,1]$ represents the probability of intrusion.

4.5 *CNN–BiLSTM Model (Deep Hybrid Model)*

A hybrid CNN–BiLSTM model improves representation learning: CNN layers capture local feature patterns, while BiLSTM models contextual dependencies, enabling richer hierarchical feature representations [20].

4.5.1 Convolutional Feature Extraction

The one-dimensional convolution operation is defined as:

$$z_k = \sigma\left(\sum_{c=1}^{C} \mathbf{x}_c * \mathbf{w}_{k,c} + b_k\right) \tag{10}$$

where $\mathbf{w}_{k,c}$ denotes the convolutional kernel, b_k is the bias term, and $\sigma(\cdot)$ is the ReLU activation function.

4.5.2 Bidirectional LSTM Layer

The LSTM unit is governed by:

$$\begin{aligned} f_t &= \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right), \\ i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i), \\ \tilde{c}_t &= \tanh(W_c x_t + U_c h_{t-1} + b_c), \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \\ o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o), \\ h_t &= o_t \odot \tanh(c_t) \end{aligned} \tag{11}$$

A bidirectional configuration processes the input in both directions:

$$h_t^{bi} = [h_t^{\rightarrow}; h_t^{\leftarrow}], \tag{12}$$

allowing the model to exploit contextual relationships across the entire feature space.

4.5.3 Output Layer

The final prediction is obtained using:

$$\hat{y} = \sigma\left(W_o h_T^{bi} + b_o\right) \tag{13}$$

where $\hat{y}$ denotes the probability of an intrusion.

4.6 *Loss Function*

Both deep learning models (MLP and CNN–BiLSTM) are trained using binary cross-entropy loss:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{14}$$

4.7 *Evaluation Metrics*

Model performance is evaluated using the following metrics:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{15}$$

$$\text{Mean Square Error (MSE)} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{16}$$

$$\text{Root Mean Square Error (RMSE)} = \sqrt{\text{MSE}} \tag{17}$$

$$\text{Mean Absolute Error (MAE)} = \frac{1}{N}\sum_{i=1}^{N} | y_i - \hat{y}_i | \tag{18}$$

4.8 *Explainability Using SHAP*

SHAP (TreeSHAP) is applied to Random Forest for exact, stable feature importance, ensuring reliable explainability, unlike approximate, costly SHAP for neural networks, which prioritize high detection performance.

5. Results and Discussion

This section evaluates Random Forest, CNN–BiLSTM, and attention-based MLP on NSL-KDD using classification metrics, confusion matrices, ROC–AUC, and SHAP for comprehensive intrusion detection analysis.

5.1 *Overall Performance Comparison*

Table 1 reports accuracy, MSE, RMSE, and MAE for all models. Random Forest achieved 77.23% accuracy with relatively higher errors. CNN–BiLSTM obtained the highest accuracy (77.84%) and lower error metrics, indicating better spatial–temporal learning. The attention-based MLP achieved 75.49% accuracy under the NSL-KDD zero-day setting. Overall, the hybrid model shows improved error reduction and discrimination over the baseline.

Table 1

Quantitative Performance of all Three Models

Model	Accuracy	MSE	RMSE	MAE
Random Forest	77%	0.2277	0.4772	0.2277
CNN-BiLSTM	78%	0.2216	0.4707	0.2216
Attention-MLP	75%	0.2450763	0.49505181	0.2450763

5.2 *CNN–BiLSTM and Attention-Based MLP Classification Analysis*

Table 2 and Figure 1 present CNN–BiLSTM and attention-based MLP results. CNN–BiLSTM achieves 0.97 precision and 0.63 attack recall, while MLP records 0.64 precision and 0.97 recall on NSL-KDD KDDTest+. Normal recall remains high (0.98 vs. 0.97). CNN–BiLSTM performs slightly better with 78% accuracy and 0.78 F1 compared to 0.75 F1 for MLP.

Table 2

Classification Report for CNN-BiLSTM and Attention-Based MLP

CNN-BiLSTM	precision	recall	f1-score
0	0.67	0.98	0.79
1	0.97	0.63	0.76
Attention-Based MLP			
0	0.64	0.97	0.77
1	0.97	0.58	0.73

5.3 *ROC–AUC Analysis*

ROC analysis evaluates performance independent of decision thresholds. CNN–BiLSTM achieves AUC = 0.8608, while the attention-based MLP reaches AUC = 0.9076 despite slightly lower accuracy as shown in Figure 2. In the imbalanced NSL-KDD zero-day setting, higher AUC indicates more stable discrimination for IDS deployment.

5.4 *Attention-Based MLP Performance Interpretation*

Despite slightly lower accuracy, the attention-based MLP achieves higher ROC–AUC (0.9076), indicating better generalization via attention-driven features. Stable errors (MSE 0.2451, RMSE 0.4951, MAE 0.2451) support reliable IDS anomaly ranking.

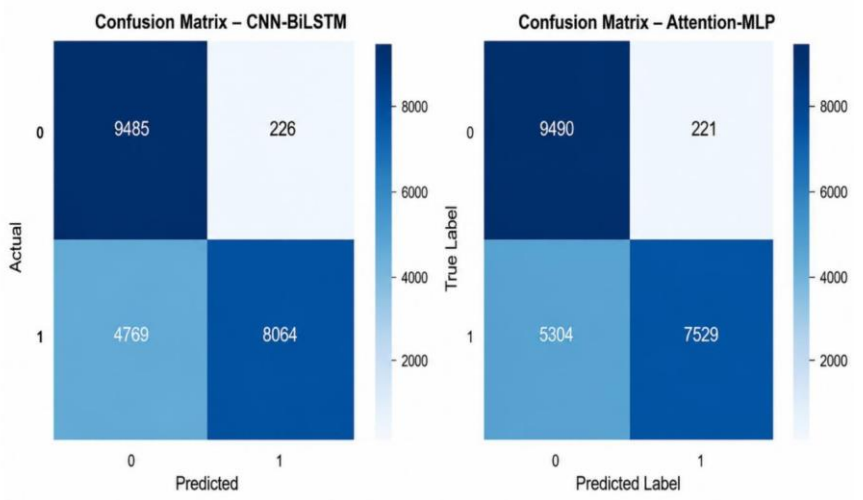

Figure 1
Confusion matric for CNN-BiLSTM and Attention MLP

5.5 *Explainability Analysis Using SHAP*

SHAP is applied to the Random Forest to obtain stable feature importance (Table 3, Figure 3). Key indicators include **src_bytes** and **dst_bytes**, highlighting abnormal traffic volume as a primary intrusion signal. Protocol and host-based features (flag, protocol_type, logged_in, dst_host_same_srv_rate, dst_host_srv_count) also significantly influence predictions.

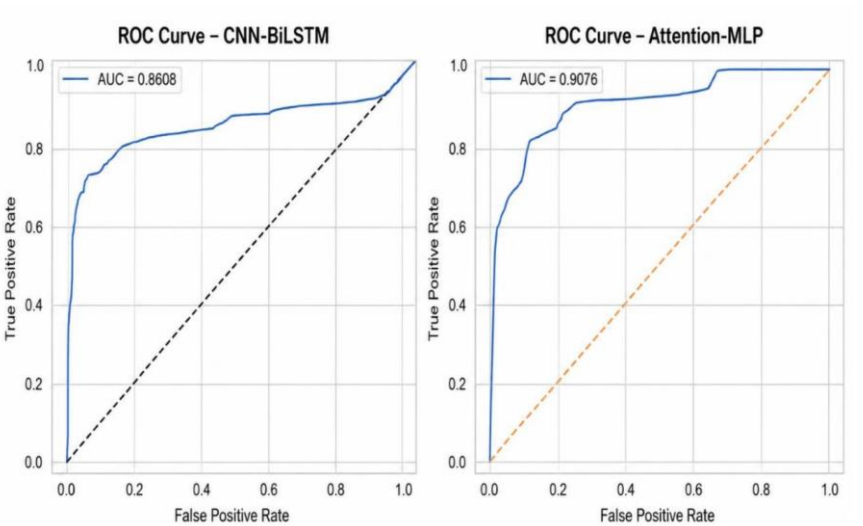

Figure 2
ROC Curve for CNN-BiLSTM and Attention MLP

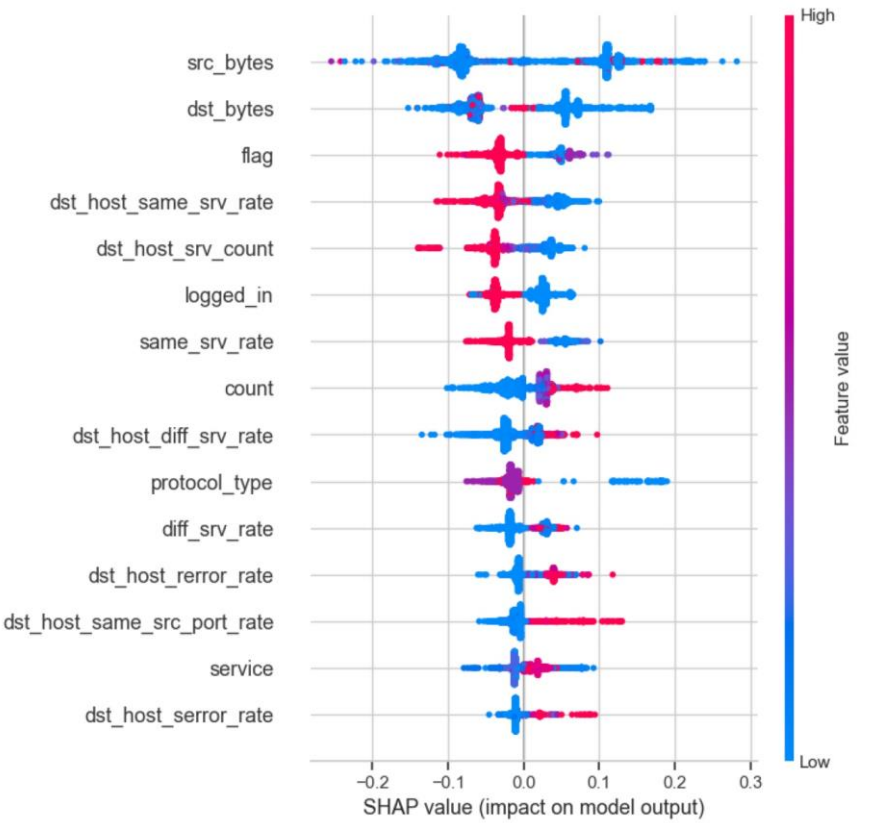

Figure 3

Top-10 features mean absolute value of NSL-KDD dataset by SHAP

5.6 *Discussion and Practical Implications*

Accuracy alone is inadequate for IDS evaluation. CNN–BiLSTM achieves the highest accuracy, the attention-based MLP shows stronger ROC–AUC ranking, and Random Forest provides SHAP-based interpretability. The 70–80% accuracy reflects the imbalanced NSL-KDD setting. Combining multiple models, evaluation metrics, and explainability creates a robust IDS framework and strengthens network defense strategies [23].

Table 3

Top-10 features mean absolute value of NSL-KDD dataset by SHAP

Feature	Mean_Absolute_SHAP_Value
src_bytes	0.09861012
dst_bytes	0.07026382
flag	0.03975828
dst_host_same_srv_rate	0.03760904

Contd...

dst_host_srv_count	0.03466709
logged_in	0.03088819
same_srv_rate	0.03060301
count	0.02637384
dst_host_diff_srv_rate	0.02502235
protocol_type	0.02282233

Conclusion

This study evaluates intrusion detection on NSL-KDD using a hybrid ML–DL framework under zero-day conditions. Random Forest, attention-based MLP, and CNN–BiLSTM were analyzed. CNN–BiLSTM achieved the highest accuracy, while the attention-based MLP showed stronger ROC–AUC. Random Forest provided baseline performance with SHAP-based interpretability, highlighting key features such as src_bytes, dst_bytes, and protocol behavior. Overall, combining deep learning with interpretable ML supports a practical IDS framework, with future work targeting multi-class detection and transformer models.

References

[1] W. Lee and S. J. Stolfo, "A framework for constructing features and models for intrusion detection systems," *ACM Trans. Inf. Syst. Secur.*, vol. 3, no. 4, pp. 227–261 (Nov. 2000), doi: 10.1145/382912.382914.

[2] D. E. Denning, "An intrusion-detection model," I*EEE Trans. Softw. Eng.*, vol. SE-13, no. 2, pp. 222–232 (Feb. 1987), doi: 10.1109/TSE.1987.232894.

[3] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Proc. IEEE Symp. Security Privacy (S&P)*, Berkeley, CA, USA, pp. 305–316 (2010), doi: 10.1109/SP.2010.25.

[4] S. Axelsson, "Intrusion detection systems: A survey and taxonomy," Technical Report No. 99-15, Dept. Comput. Eng., Chalmers Univ. Technol., Göteborg, Sweden, (Mar. 2000).

[5] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844 (Aug. 1998), doi: 10.1109/34.709601.

[6] G. Kim, S. Lee, and S. Kim, "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 1690–1700 (2014).

[7] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961 (2017), doi: 10.1109/ACCESS.2017.2762418.

[8] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, "Long short-term memory recurrent neural network classifier for intrusion detection," in *Proc. 15th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Anaheim, CA, USA, pp. 471–476 (Dec. 2016), doi: 10.1109/ICMLA.2016.0086.

[9] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 1, pp. 41–50 (Feb. 2018), doi: 10.1109/TETCI.2017.2772792.

[10] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?: Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, San Francisco, CA, USA, pp. 1135–1144 (2016), doi: 10.1145/2939672.2939778.

[11] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30 (2017).

[12] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Security Defense Appl. (CISDA)*, Ottawa, ON, Canada, pp. 1–6 (2009), doi: 10.1109/CISDA.2009.5356528.

[13] D. Gaspar, P. Silva, and C. Silva, "Explainable AI for intrusion detection systems: LIME and SHAP applicability on multi-layer perceptron," *IEEE Access*, vol. 12, pp. 30164–30175 (2024), doi: 10.1109/ACCESS.2024.3369874.

[14] P. Ramyavarshini, G. K. Sriram, U. Rajasekaran, and A. Malini, "Explainable AI for intrusion detection systems," in *Proc. 5th Int. Conf. Contemporary Computing and Informatics (IC3I)*, pp. 1563–1567 (2022), doi: 10.1109/IC3I56241.2022.10072682.

[15] H. Hindy, D. Brosset, E. Bayne, A. K. Seeam, X. Bellekens, C. Tachtatzis, and R. Atkinson, "A taxonomy of network threats and the effect of deep learning in cybersecurity," *IEEE Access*, vol. 8, pp. 97433–97488 (2020), doi: 10.1109/ACCESS.2020.2995333.

[16] A. Y. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A Deep Learning Approach for Network Intrusion Detection System," *EAI Endorsed*

Transactions on Security and Safety, vol. 3, no. 9, p. e2 (May 2016), doi: 10.4108/eai.3-12-2015.2262516.

[17] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. 2009 IEEE Symp. Comput. Intell. Security Defense Appl. (CISDA),* Ottawa, ON, Canada, pp. 1–6 (Jul. 2009), doi: 10.1109/CISDA.2009.5356528.

[18] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, USA: Springer (2006).

[19] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, USA: Springer (2009).

[20] W. Lee and S. J. Stolfo, "A framework for constructing features and models for intrusion detection systems," *ACM Trans. Inf. Syst. Secur.,* vol. 3, no. 4, pp. 227–261 (Nov. 2000).

[21] V. P. Sharma, N. S. Yadav, S. S. Adavi, D. S. D. Reddy, and B. B. Gupta, "A two stage hybrid intrusion detection using genetic algorithm in IoT networks," *Journal of Discrete Mathematical Sciences and Cryptography,* vol. 26, no. 3, pp. 667–676 (2023).

[22] S. K. Henge, A. Upadhyay, A. K. Saini, N. Mishra, D. Sharma, and G. Sharma, "Analysis and detection of insider attacks using behaviour rule-based architecture in enterprise multitenancy," *Journal of Discrete Mathematical Sciences and Cryptography,* vol. 26, no. 3, pp. 707–718 (2023).

[23] P. Vats, S. K. Vats, and P. Peddi, "Unveiling crypto analysis secrets: A comprehensive analysis of smart contract security within blockchain network environments," *Journal of Discrete Mathematical Sciences and Cryptography,* vol. 27, no. 4, pp. 1121–1128 (2024).

Received December, 2025