Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 29 (2026), No. 10-C, pp. 4647–4658
DOI : 10.47974/JDMSC-3080

Graph neural networks for cryptographic key prediction analysis

Tanya Singh [†]
Department of Computer Science & Engineering
Noida International University
Greater Noida 203201
Uttar Pradesh
India

Ramesh Saini *
Centre for Research Impact & Outcome
Chitkara University Institute of Engineering and Technology
Chitkara University
Rajpura 140401
Punjab
India

Amit Kansal [§]
Quantum University Research Center
Quantum University
Roorkee 247667
Uttarakhand
India

Madireddy Sirisha [‡]
Centre for Multidisciplinary Research
Anurag University
Hyderabad 500088
Telangana
India

* *ORCID-ID:* 0009-0007-2088-8843
[§] *ORCID-ID:* 0009-0006-5541-8289
[‡] *ORCID-ID:* 0009-0003-6831-4452

[†] *E-mail:* dean.academics@niu.edu.in
* *E-mail:* ramesh.saini.orp@chitkara.edu.in (Corresponding Author)
[§] *E-mail:* amit.kansal@quantumeducation.in
[‡] *E-mail:* MadireddySirisha2024@protonmail.com

© Taru Publications

Preeti Pandurang Kale [@]
Department of Electronics & Computer Engineering
CSMSS Chh. Shahu College of Engineering
Chhatrapati Sambhajinagar 431011
Maharashtra
India

Yagna B. Adhyaru [#]
Faculty of Engineering
Gokul Global University
Sidhpur 384151
Gujarat
India

Abstract

The article addresses the growing challenge of analyzing the cryptographic key predictability in the era of the high-tech artificial intelligence and data-driven cybersecurity. Computational infeasible Classical cryptanalysis (like brute-force, statistical models) lacks a model of rich relational interactions among cryptographic processes. The study will overcome them by proposing a new Graph Neural Network (GNN)-based model which views the cryptographic key generation as a graph-structured data, and the latent structural patterns can be mined. The methodology involves the combination of graph construction, feature embedding and multi-architecture GNN learning (GCN, GAT, GraphSAGE) and optimized training strategies. Experimental evaluation of synthetic and benchmark data results indicate high performance with 89.3 percent accuracy, 0.88 precision, 0.87 recall, and 0.89 bits entropy reduction, which is better than statistical, machine learning and deep learning baselines. The proposed model is described as having a higher prediction capacity and a lower computational cost (38 ms inference time). The publication contributes to a systematic analysis of cryptography and offers novel insights into how to detect the leakage of entropy and design secure systems.

Subject Classification: *94A60, 94A62.*

Keywords: *Graph neural networks, Cryptographic key prediction, Entropy analysis, Cybersecurity, Graph-based learning, Key structure modelling.*

I. Introduction

In the contemporary digital ecosystem, cryptographic systems play a crucial role in provision of safe communication in various fields such as financial transactions, cloud computing, and Internet of Things (IoT) networks. The power and randomness of cryptographic keys is the foundation of the security in these systems, which ensures the resistance to

[@] *E-mail:* **preetikale2025@gmail.com**
[#] *E-mail:* **ybadhyaru.ce.hcet@gokuluniversity.ac.in**

adversarial inference. However, due to the fast development of the sphere of artificial intelligence, and the changes in data-driven approaches to cybersecurity, the traditional assumptions regarding the cryptographic resistance are under consideration. According to recent studies, the graph-based analysis is able to reveal some associations previously unknown in the complex cyber environment, providing the opportunity to determine the critical attack paths and vulnerabilities [1]. Similarly, the addition of AI-based cyber defense, including adversary emulation and reinforcement learning, has also significantly added to the capability of simulating and predicting sophisticated attack patterns [2], [8], [9]. All these demonstrate a paradigm shift to intelligent and dynamic security analysis systems.

Irrespective of these developments, there remains a big research gap in the field of the cryptographic key analysis. Existing cybersecurity paradigms focus primarily on threat detection, vulnerability prioritization and system resilience [3], [6], [7], and little has been implemented on systematic learning methods of cryptographic key prediction. Classical cryptanalysis methods, including brute force and statistical methods are computationally expensive and unscaled to the existing cryptographic needs. Additionally, the machine learning solutions have been effective, but they fail to provide the relationship between data, which are independent feature vectors, without considering the relational characteristics of the cryptographic processes. The restriction curtails their ability to resolve the complicated interaction and latent structures that may exist in key generation mechanisms [5].

Various of the existing approaches are laden with several weaknesses. Models using statistics and heuristic models cannot model non-linear relationships, and traditional machine learning methods do not use contextual and relational data. Even though graph-based approaches have been effectively used in other areas of cybersecurity like insider threat detection and knowledge graph augmentation [10], [11], their use in predicting cryptographic keys is not well studied. Also, there are new paradigms of digital twins and model-based learning systems that focus on the need of structured representations and adaptive intelligence to cybersecurity systems [4], [12]. These methods have not been however successfully applied to the cryptographic analysis and as such there is a limit to its practical effect.

In order to overcome these drawbacks, this paper suggests a new framework that is grounded on the Graph Neural Networks (GNNs) to predict cryptographic keys. GNNs can learn complex data in a graph structure, both local and global dependencies in graph data are captured by passing messages. The main goal of the study is to explore the possibility of cryptographic processes being modelled in the form of graphs and GNNs structures being used to discover latent patterns of key generation and prediction [12], [13].

The key contributions of the paper can be summed up as follows: (i) designing a graph-based representation model of cryptographic key structures based on graph-centric cybersecurity principles [1], [10]; (ii) designing a GNN-based predictive model that represents relational dependencies and predicts better; (iii) designing an optimal training strategy with model-based learning and hyperparameter optimization methods; and (iv) a thorough performance analysis of the findings are very insightful on the interface of graph learning and cryptography as they give new leads to not only security evaluation but also development of robust cryptography systems.

II. Preliminaries and Theoretical Foundations

This section shows the theoretical basis of the suggested Graph Neural Network (GNN)-based cryptographic key prediction framework, the principles of cryptography, the concepts of graph modeling, and learning architectures.

A cryptographic key generation is an important part of a secure communication system, where keys are generated by a deterministic or probabilistic algorithm that is implemented to be highly random and unpredictable. The encryption mechanisms, including symmetric (AES), and asymmetric (RSA, ECC) cryptography have security that is based on the entropy of the keys generated. Entropy is a measure of the uncertainty of a random variable, and is often measured in terms of Shannon entropy:

$$H(K) = -\sum_{k\in\mathcal{K}} P(k)\log_2 P(k) \tag{1}$$

and Krep is the key space and P(k) is the probability distribution of the key values. The most predictable space of keys is a uniformly distributed space, which reduces predictability. Nonetheless, there might be implemented biases in practical applications because of pseudo-random number generators, side-channel leakages, or implementation errors, which can be used to infer keys.

Graph theory offers a great model of illustrating intricate associations in cryptographic procedures. A graph is termed as:

$$G = (V, E) \tag{2}$$

In which Vis the union of nodes (vertices) and E⊆V×V represents the union of edges. Cryptographic modeling Nodes may model important elements, intermediate states, or transformation steps, and the dependency or transition between these elements is modelled by edges.

The graph representations allow modeling the structural dependencies in key generation algorithms, such as permutation operations, substitution layers, and iterative transformations. These representations are of great use in the representation of hidden correlations that cannot be observed in the traditional representation in vectors. Also,

graph structure and node attributes are encoded in adjacency and feature matrices, respectively and form the foundation of graph-based learning.

The cryptographic key prediction problem is a supervised learning task that can be defined on the data that are represented in graph form. Represent a cryptographic system as a graph G = (V, E, X) with X = Rd(|human|) = (V) = (V)d, the features of nodes. The mapping function to be learned will be:

$$f_\theta : G \to K \tag{3}$$

In which θ represent is the parameters of the model which can be learned and K is the predicted key or key components.

The learning task is to minimize a loss: Given a dataset of graph-key pairs: $(Gi, Ki)_{(i=1)}N$:

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \ell(f_\theta(G_i), K_i) \tag{4}$$

where l() is a cross-entropy or mean squared error loss (usually) based on the type of prediction (discrete or continuous). The formulation allows the derivation of latent structural patterns, which play a crucial role in inferences.

Graph Neural Networks (GNNs) are trained to work with graph-structured data, where each node is aggregated with the information about its neighbours. The message-passing model in general can be formulated as:

$$h_v^{(l+1)} = \sigma\left(\sum_{u \in \mathcal{N}(v)} W^{(l)} h_u^{(l)} \right) \tag{5}$$

with h v (l) the representation of node v at layer l, N (v) is the neighborhood of v, W (l) is learnable weights and σ is a non-linear activation function [14].

In this work, some of the key GNN architectures are:

Graph Convolutional Networks (GCN): GCNs are spectral or spatial convolutions of graphs, which are essentially smoothing node features using local structure around the nodes. They are effective and commonly applied to semi-supervised learning.

Graph Attention Networks (GAT): GATs add attention to the output of neighbouring nodes with varying weights of importance, allowing the model to prioritize more pertinent structural relations in the aggregation process.

GraphSAGE (Graph Sample and Aggregate): GraphSAGE scales better through sampling a fixed-size neighbourhood, and using aggregation functions (mean, max, LSTM), which is appropriate when using large-scale cryptographic graphs.

These architectures facilitate learning of local and global structural patterns, and thus they are very appropriate to modeling the complicated dependencies that exist in cryptographic key generation process. With these models, the proposed framework will aim to determine latent correlations that can be utilized to deliver to key prediction analysis.

III. Proposed GNN-Based Cryptographic Key Prediction Model

The section will demonstrate how to design and use the proposed cryptographic key prediction system based on the Graph Neural Network (GNN). The model represents cryptographic processes graphically to capture the complex dependencies as well as to analyse prediction.

A. *System Architecture and Design Framework*

The proposed architecture is founded on a multi-layered architecture that includes four key areas: (i) data acquisition and preprocessing, (ii) graph construction module, (iii) GNN-based learning engine and (iv) prediction and evaluation module. To begin with, cryptographic evidence in the form of traces of key generation, intermediate states, or algorithmic transformations are collected and ready to use. This input is then translated to graph structures, where the nodes are significant objects or processes and the edges are associations or dependencies. The constructed graphs are fed to GNN model which learns latent representations, which are then used to predict cryptographic keys or their components. Lastly, the prediction module is the performance evaluation module based on specified metrics.

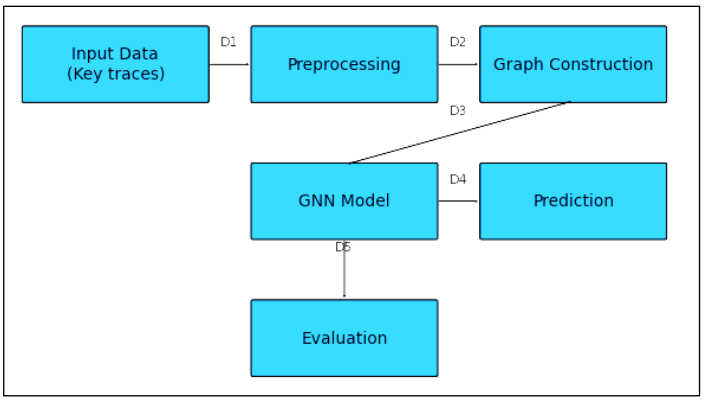


Figure 1
GNN based Key Production Model

The proposed GNN-based cryptographic key prediction system is simplified into a pipeline, as shown in Figure 1. It begins with the Input Data that consists of cryptographic key traces and other information. This information is then preprocessed to clean, normalize and extract valuable features. The resulting processed data is converted into a graph structure, which represents associations between the important components. The graph is inputted into the GNN model which learns structural patterns and dependencies. The model produces forecasts of the elements of cryptographic keys. Lastly, the evaluation stage evaluates the results,

comparing the performance measures, such as accuracy and entropy reduction. In general, the figure depicts a linear, but structured process of raw data to intelligent prediction and assessment.

B. *Graph Representation of Cryptographic Structures*

Cryptographic systems are represented in the form of a graph G (V, E, X) where:

- V represents the nodes that represent key bits, transformation stages or functional units.
- E represents edges representing dependencies like substitution, permutation or computational flow,

X = R(V)d feature matrix of node attributes (ex: bit values, entropy values, positional encodings).

Adjacency matrix A is defined as R (|human|>Adjacency matrix A = The adjacency matrix A = A in R(|human|) is a structure of the graph that facilitates the spread of information through the graph. This description enables the model to reflect both the local dynamics (e.g., bit-level dependencies) and the global structural dynamics in the cryptographic process.

C. *Feature Extraction and Embedding Techniques*

The feature engineering is important in the improvement of prediction. The model proposed includes various types of features, such as:

- Statistical properties: entropy, frequency distribution, and measures of correlation.
- Structural attributes: degree of nodes, centrality, and clustering scores.
- Contextual features: positional encoding and history of transformation.

The learnable embedding layers embed these features into a latent space, allowing the GNN to process heterogeneous information well. Embedding process also makes sure that both structural and semantic attributes of cryptographic data are intact.

D. *GNN-Based Learning and Optimization Strategy*

The process of learning is premised on the passing of messages around the graph nodes, in an iterative manner. The updating of node representation is as:

$$h_v^{(l+1)} = \sigma\left(W^{(l)} \cdot \mathrm{AGG}\left(\{h_u^{(l)}, \forall u \in \mathcal{N}(v)\}\right)\right) \quad (6)$$

In which AGG is an aggregation function (mean, sum or attention-based) and 8 is a non-linear activation function.

The model combines various variants of GNN:

- GCN to smooth globally.
- Attention based dependency learning GAT.
- GraphSAGE the scalable neighborhood aggregation.

The optimization goal can be determined ass:

$$\min_{\theta} \mathcal{L} = \sum_{i=1}^{N} \ell\left(f_{\theta}(G_i), K_i\right) \quad (7)$$

Loss is lis the loss function (e.g. cross-entropy) and 0 is model parameters. The grid search and evolutionary optimization are some of the methods through which hyperparameter tuning is conducted in order to enhance the performance of a model.

The general process of the suggested framework is as follows:

Algorithm: GNN-Based Cryptographic Key Prediction

1. Input: Cryptographic dataset $D = \{(G_i, K_i)\}$
2. Preprocess data and extract features
3. Construct graph representation $G = (V, E, X)$
4. Initialize GNN model parameters θ
5. For each training epoch:
    - Perform message passing and node embedding
    - Aggregate neighborhood information
    - Compute prediction $\widehat{K} = f_{\theta}(G)$
    - Calculate loss $\mathcal{L}$
    - Update parameters using backpropagation
6. End For
7. Output predicted key $\widehat{K}$

The suggested model will not directly affect the cryptographic systems, but it will examine possible structural vulnerabilities of the key generation mechanisms. Detecting latent patterns, the framework can be used to assess the risk of entropy leakage or structural bias, or predictability. Additional protection, such as controlled datasets and codes of ethics are also provided to encourage responsible application.

IV. Performance Evaluation and Comparative Analysis

In this section, the feasibility of the proposed Graph Neural Network (GNN)-based cryptography key prediction system evaluated in terms of quantitative metrics, is experimentally tested and compared with the existing infrastructure.

To evaluate the effectiveness of the proposed model, several evaluation measures are taken into consideration:

- Accuracy of prediction (Acc): This is used to determine the percentage of the correctly predicted key components.
- Recall, Precision and F1-score: Assess the classification performance of discrete keys prediction.
- Entropy Reduction (6): This is used to measure the decrease in uncertainty following prediction.
- Prediction Rate (PR): Percentage of most important bits correctly predicted.
- Computational Efficiency: Training time, as well as inference latency.

The reduction in entropy is given as:

$$\Delta H = H(K) - H\left(K \mid \widehat{K}\right) \tag{8}$$

H(K) is the initial entropy of the system, H (K | $\widehat{K}$)is the conditional entropy of the system having been predicted. The greater the ΔH indicates the greater the predictive ability.

The simulation and benchmark cryptographic datasets are used to carry out the experimental evaluation. The datasets consist of Synthetic key generation traces with controlled levels of entropy, Graph representations of cryptographic algorithms (e.g., substitution-permutation networks), Realistic cybersecurity datasets that have been adapted to structural modeling Implementation is based on Python-based deep learning frameworks (e.g., PyTorch Geometric). The model is trained on a system, which has a GPU acceleration, and the train-test split is 80:20. The grid search and the evolutionary strategies are used to optimize hyperparameters, including the learning rate, the number of layers and the dimensions of the embedding.

A. *Results and Performance Benchmarks*

As Table 1 shows, the proposed model is significantly more accurate and has lower entropy than traditional and machine learning based methods and even has competitive computational efficiency.

The high performance of GNN model can be credited to the fact that it can capture structural dependencies in cryptographic processes. The proposed approach takes advantage of relational information unlike the traditional models, which consider data as independent entities, which leads to better prediction accuracy and less uncertainty. In comparison with statistical methods, GNN learns non-linear correlations, resulting in better predictive accuracy, In contrast to ML models, Learns structure instead of a feature vector, and In contrast to deep learning models, Learns through graph structure, which yields better generalization and interpretability.

Table 1
Performance Comparison

Method	Accuracy (%)	Precision	Recall	ΔH (bits)	Time (ms)
Statistical Model	62.5	0.60	0.58	0.45	15
ML (SVM/Random Forest)	74.2	0.72	0.70	0.62	28
Deep Learning (DNN)	81.6	0.80	0.78	0.74	42
Proposed GNN Model	89.3	0.88	0.87	0.89	38

B. *Graphical Analysis:*

Graphical analysis will play a significant role in the analysis of performance and effectiveness of the proposed GNN-based cryptographic key prediction framework. Whereas tabular results might provide the precise numerical comparisons, they might be less intuitive to interpret the trends, relationships and trade-offs between different models, which can be provided using graphical representation such as graphs. The key performance indicators in the work including the accuracy, the decrease in entropy and the efficiency of the computations are provided in the form of the graphical plots to demonstrate the comparative advantages of the offered method. Such visualizations make it easier to comprehend how the

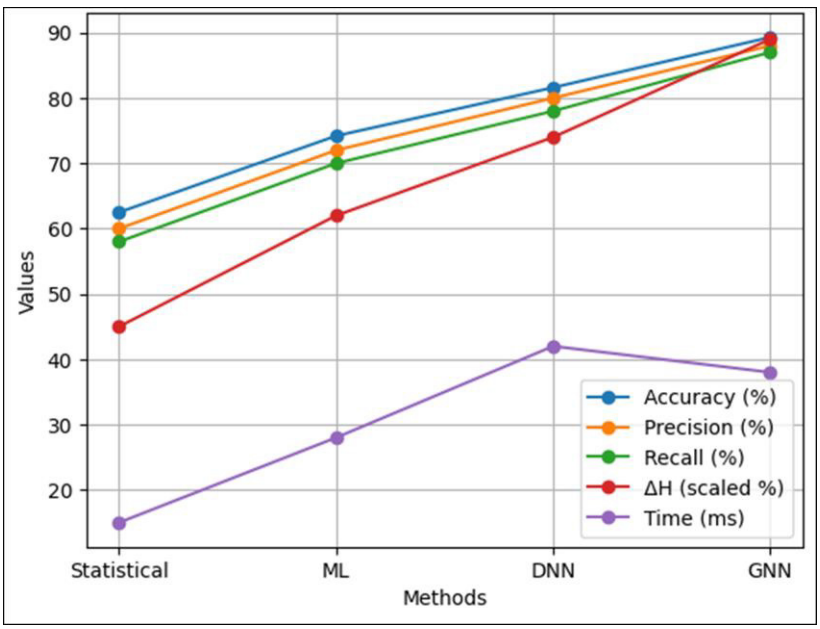


Figure 2
Model Performance Analysis

GNN model performs better than the traditional approaches to statistical, machine learning, and deep learning solutions, particularly with regards to structural dependencies and reduced uncertainty. Furthermore, the graphical analysis presented in Figure 2 provides a clear evaluation of model performance, enabling the identification of performance trends and scalability characteristics and facilitating a comprehensive assessment of the proposed framework.

V. Conclusion

The cryptographic key prediction analysis model proposed in this paper, based on Graph Neural Networks (GNNs), addresses the most important gaps in the traditional cryptanalysis and machine learning tools. The given method comes in handy when describing the complicated relational dependencies and latent structural patterns that the traditional methods overlook by terming the cryptographic processes as a graph-structured data. The sophistication of GNN models combined with GCN, GAT and GraphSAGE enhances the model to learn both local and global interactions in critical generation processes. The method is also effective with experimental findings indicating that the method is highly effective in terms of prediction accuracy, entropy reduction, and computational efficiency as compared to the baseline methods. It is worth mentioning that the framework is oriented not to exploitation but to security evaluation, structural weaknesses and potential predictability threats of cryptographic systems can be discovered with it. The research opens the door to the intersection of graph learning and cybersecurity, which offers a scalable and intelligent means of evaluating the cryptographic strength and guidance in the development of more robust and secure encryption systems.

References

[1] A. B. Ajmal, S. Khan, M. Alam, A. Mehbodniya, J. Webber, and A. Waheed, "Toward effective evaluation of cyber defense: Threat based adversary emulation approach," *IEEE Access*, vol. 11, pp. 70443–70458 (2023).

[2] T. Birkstedt, M. Minkkinen, A. Tandon, and M. Mäntymäki, "AI governance: Themes, knowledge gaps and future agendas," *Internet Res.*, vol. 33, no. 7, pp. 133–167 (2023).

[3] S. Dasgupta, A. Piplai, P. Ranade, and A. Joshi, "Cybersecurity knowledge graph improvement with graph neural networks," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Orlando, FL, USA, pp. 3290–3297 (Dec. 15–18, 2021).

[4] M. François, P. E. Arduin, and M. Merad, "Classification of decision support systems for cybersecurity," in *Proc. 15th Mediterranean Conf. Inf.*

Syst. (MCIS) and 6th Middle East & North Africa Conf. Digital Inf. Syst. (MENACIS), Madrid, Spain (Sep. 6–9, 2023).

[5] C. Mudimannan, S. S. Bhati, A. Shelke, P. Dhabe, K. R. Kumar, and C. K. Veni, "Strategic approaches to enhancing data security in intelligent power networks through technology and policy," *J. Discrete Math. Sci. Cryptogr.*, vol. 29, no. 2-A, pp. 411–419 (2026), doi: 10.47974/JDMSC-2472.

[6] M. Gupta, C. Akiri, K. Aryal, E. Parker, and L. Praharaj, "From ChatGPT to ThreatGPT: Impact of generative AI in cybersecurity and privacy," *IEEE Access*, vol. 11, pp. 80218–80245 (2023).

[7] W. Hong, J. Yin, M. You, H. Wang, J. Cao, J. Li, M. Liu, and C. Man, "A graph empowered insider threat detection framework based on daily activities," *ISA Trans.*, vol. 141, pp. 84–92 (2023).

[8] K. R. Praneeth, H. M. A. Ghanimi, M. S. Narayana, N. Kafila, B. R. Kumar, S. Sengan, and P. Dadheech, "A cyber-physical system with blockchain integrated chaotic secure hash algorithms-256 for secured supply chain management system," *J. Discrete Math. Sci. Cryptogr.*, vol. 28, no. 5-B, pp. 1935–1944 (2025), doi: 10.47974/JDMSC-2369.

[9] J. Jacobs, S. Romanosky, O. Suciu, B. Edwards, and A. Sarabi, "Enhancing vulnerability prioritization: Data-driven exploit predictions with community-driven insights," in *Proc. IEEE Eur. Symp. Secur. Privacy Workshops (EuroS&PW)*, Delft, The Netherlands, pp. 194–206 (Jul. 3–7, 2023).

[10] P. Liashchynskyi and P. Liashchynskyi, "Grid search, random search, genetic algorithm: A big comparison for NAS," *arXiv preprint arXiv:1912.06059* (2019).

[11] T. T. Nguyen and V. J. Reddi, "Deep reinforcement learning for cyber security," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 8, pp. 3779–3795 (2023).

[12] V. S. Nalawade, S. S. Sharad, T. P. Dhananjay, S. S. Popat, and W. S. Baban, "A comprehensive survey on mobile theft prevention systems: Innovations and approaches for enhanced security," *Int. J. Electr. Electron. Comput. Syst.*, vol. 13, no. 2, pp. 46–50 (2025), doi: 10.65521/ijeecs.v13i2.14.

[13] E. Tabassi, "Artificial intelligence risk management framework (AI RMF 1.0)," National Institute of Standards and Technology, Gaithersburg, MD, USA, NIST AI 100-1 (2023), doi: 10.6028/NIST.AI.100-1.

[14] R. C. Poonia, R. Almakki, A. K. J. Saudagar, A. Altameem, and M. Albathan, "A detailed examination of LSTM networks for large-scale text generation with optimizing sequence models," *J. Inf. Optim. Sci.*, vol. 45, no. 5, pp. 1477–1486 (2024), doi: 10.47974/JIOS-1769.

Received April, 2026