Journal of Intelligent Systems and Internet of Things
Vol. 18, No. 02, PP. 20-35, 2026
DOI: https://doi.org/10.54216/JISIoT.180202

A New Descriptor for Improving Lightweight Blockchain Environment Using a Hybrid GWO-Levy-GRU Framework for Nonce Discovery

Rasha Hani Salman[1,*], Hala Bahjat Abdul Wahab[2]
[1]Informatics Institute for Postgraduate studies, Information Technology & Communication University, Baghdad, Iraq
[2]Computer Sciences Department, University of Technology, Baghdad, Iraq
Emails: rsalman@uowasit.edu.iq; Hala.B.AbdulWahab@uotechology.edu.iq

Abstract

Blockchain technology has recently emerged as a fundamental pillar of decentralized and secure systems. However, many Proof-of-Work (POW) algorithms suffer from some challenges, including their inefficiency in discovering the value of Nonces due to their reliance on random attempts, which consume significant resources, energy, and time, making them difficult to use in lightweight blockchain environments, especially in resource-limited environments such as mobile devices and others. The main goal of this paper is to introduce a smart system that replaces random guessing with a more intelligent and predictive approach using deep learning models like CNN2D, GRU, LSTM, and hybrid models. The intelligent optimization algorithm (GWO) is also used, which has been enhanced with random Lévy jumps, in addition to improved clustering using a genetic algorithm. The results, after applying the system to health data across three difficulty levels (4, 6, and 8), showed that the intelligent neural model was the most stable and accurate, achieving the lowest error values and the highest generalization ability, with a maximum error value of (0.0136) at the highest difficulty level (8). The hybrid GA–KMeans algorithm demonstrated high efficiency in improving clustering accuracy. It achieved the highest similarity index value (0.9980) and the lowest Davis-Bolden index value (0.0000), which plays a significant role in guiding searches efficiently and effectively. The CNN2D model also achieved ideal numerical results, but it is prone to overlearning, while the GRU neural model provided an efficient balance between stability and accuracy. Other hybrid models, such as GRU+CNN, have shown excellent performance, but with varying results. The proposed system proves to be an efficient and intelligent alternative to the low-cost random approach for Nonce discovery in lightweight blockchain environments.

Received: March 03, 2025 Revised: June 02, 2025 Accepted: July 16, 2025

Keywords: Neural Network; Grey Wolf Algorithm; Levy Flight; Lightweight Blockchain Systems; Ascon Algorithm; Smart Mining; Nonce Value Optimization

1. Introduction

Recently, one of the most advanced technologies in the digital world has become widespread: blockchain. Proposed by Satoshi Nakamoto in 2008 [1, 2], this technology has undergone significant development to become the cornerstone of building secure and scalable distributed systems [3, 4]. Blockchain was originally designed to support the digital currency Bitcoin, but its fundamental structure, based on decentralization, encryption, and linked blockchains, has made it important and useful in many other fields such as healthcare, smart contracts, and the Internet of Things [5- 7]. Blockchain technologies operate according to the Proof of Work (POW) mechanism. One of the important operations on which the POW mechanism relies is searching for a numerical value called a Nonce, which allows the hash function to produce a specific pattern that satisfies the difficulty condition [8, 9]. Finding this value (Nonce) is one of the most energy- and time-consuming operations, making its improvement a primary goal of much recent and contemporary

research [10, 11]. Research has shown that most blockchain networks use hashing algorithms such as SHA-256, which are highly secure. They produce unique keys for each transaction. However, these algorithms are random in nature, making mining a significant challenge [12, 13]. To overcome these challenges, biologically inspired techniques and intelligent algorithms can be used to guide the search for the Nonce value more accurately and efficiently by balancing exploration and exploitation. The system proposed in this paper combines a deep learning neural network (GRU) algorithm with a bioinspired method, the Grey Wolf (GOW) algorithm, which is optimized using Levy Flight. This proposed model aims to make finding the Nonce value faster and more efficient in lightweight blockchain systems by intelligently evaluating different solutions and utilizing the statistical features of hash functions. The ability to address one of the most significant challenges facing Proof-of-Work (PoW) systems in lightweight blockchains—the high resource and time consumption of the optimal Nonce value—is the main thrust of this research, which seeks to make a technical and scientific contribution. This system introduces a new hybrid model that combines three effective methods: the GRU network for intelligent prediction and evaluation, the Grey Wolf (GWO) algorithm for structured group search, and Levy Flight jumps to expand the search range and avoid local solutions. Using advanced statistical features such as entropy, frequency, diversity, and random number Nonce, it integrates them into the GRU neural model to improve the accuracy of random number evaluation and better assess the quality of solutions in a smarter and more realistic way. This system contributes to the advancement of scientific research in intelligent algorithms and blockchain, providing new perspectives by combining group evolution with predictive intelligence to address critical and critical computing problems. By reducing the number of random attempts and shortening the time required to find the appropriate value for the random number, mining in a blockchain environment becomes fast and efficient. Reducing energy consumption and computing resources is critical in resource-constrained environments, such as mobile devices. The integration of the GRU model, which is capable of learning from data changes and continuously adjusting its evaluations, enhances the adaptability of blockchain networks to fluctuating difficulty levels. The proposed model can be used in various applications outside the blockchain framework, such as dynamic encryption, improving random generators, or improving the performance of mining algorithms in real-time environments. Therefore, this study presents an improved mathematical model and proposes a conceptual shift toward smart and adaptive mining systems, enhancing the efficiency and security of future blockchain systems.

This research includes a set of interconnected sections aimed at constructing an integrated intelligent model to improve the Nonce discovery process in lightweight blockchain environments; section 2 highlights the pivotal scientific contribution of this work, the intelligent integration of evolutionary optimization algorithms and deep learning models. Section3 presents the most prominent previous studies, focusing on their strengths and weaknesses. Section4 focuses on the most important knowledge gaps that have not been addressed previously, which represent the basic starting point for this research. The section5 highlights the urgent need to develop an intelligent model in the blockchain environment, especially in sensitive sectors such as healthcare, to reduce energy and time consumption without compromising performance and security. Section6 explains the methodology adopted in building the model, starting from the process of feature extraction, data collection, and the process of predicting the quality of Nonce using GRU, all the way to the final stage, which is optimization via GWO-Levy. Section7 introduces the intelligent GRU model and demonstrates its vital role in predicting the percentage of zeros as an indicator of Nonce quality. Section 8 addresses the intelligent optimization stage using a hybrid intelligent algorithm supported by multi-threaded execution. This is followed by section Nine, which provides a detailed description of the medical data used to test the model, while section ten presents the results of the evaluation and comparison between the performances of different neural models across three difficulty levels. The final chapter of this research highlights the conclusions and future recommendations that aim to enhance the employability of the proposed smart model in real-world blockchain applications such as healthcare.

The most prominent scientific contribution of this research is the design and implementation of a hybrid intelligent framework that effectively improves the process of Nonce discovery in lightweight blockchain environments, which can be summarized in the following points:

1. A deep learning-based intelligent predictive model is proposed, relying on statistical properties extracted from hash outputs, such as zero percentage, entropy, and diversity, to accurately and reliably assess the quality of Nonce filters. This represents an excellent alternative to traditional random sampling in lightweight blockchain environments, especially those based on the Proof of Work (PoW) algorithm.
2. The paper presents an improved Gray Wolf Orbiter (GWO) algorithm, inspired by the collective hunting behavior of gray wolves, augmented by a Lévy Flight mechanism to expand the search scope, increase the exploration efficiency of the solution space, and avoid localized solutions. This helps improve Nonce value discovery in blockchain environments.
3. In this research, a hybrid technique combining KMeans algorithm with genetic algorithm (GA-KMeans) is proposed to improve the accuracy of data optimization and intelligently process the search space, which in turn contributes to raising the efficiency of deep learning models and optimization algorithms for Nonce detection.
4. The research proposed an integrated system that combines the deep learning algorithm (GRU), swarm intelligence (GOW), and chaotic stochastic search reinforcement (Lévy Flight) in a new hybrid framework.

5. This research demonstrates the importance of quantitative improvements (such as maximum error) in the accuracy of random value prediction, generalization ability, and clustering quality across different difficulty levels (4, 6, 8).
6. This research contributes to filling a research gap by linking deep neural prediction with evolutionary optimization for Nonce detection, which has not been previously integrated into the same architecture.
7. This research contributes to bridging a research gap by combining deep neural prediction with evolutionary optimization for Nonce detection, an integration that has not been employed in previous studies (as will be mentioned) within the same hybrid architecture.

2. Related works

Many scientific studies have emerged in recent years that address improving the performance of blockchain systems, particularly with regard to finding the ideal Nonce number using evolutionary systems and artificial intelligence algorithms. The most important studies are as follows:

The aim of the research in [14] is to simulate the collective hunting behavior of gray wolves to develop a new evolutionary optimization algorithm known as GOW. The purpose of this study is to develop an algorithm based on the hierarchy of wolves (α, β, δ) to guide the search towards the best solutions in a dynamic manner. The benefit of this research is to develop a simple but effective computational framework between exploration and exploitation, which makes it suitable for many applications in which we build blockchain applications. In [15], the researcher discussed how to incorporate the concept of Levy Flight within evolutionary algorithms as an important means of improving random exploration and overcoming stagnation points. The primary goal of this study is to use the heavy-tailed distribution to create large computational leaps that allow the algorithm to escape from local solutions. The importance of this study lies in the effective improvement of algorithms that combine the GOW and PSO algorithms when combined with Levy Flight to reach optimal solutions.

The researcher in [16] proposed a new type of neural network known as Gated Recurrent Unit (GRU), which is a faster and lighter algorithm compared to the LSTM algorithm, and is characterized by its efficiency in dealing with sequential data. The goal of this study is to provide a network structure that can learn long-term temporal relationships without using multiple layers. The benefit of this research may be that it provides an intelligent neural model that can analyze sequential or temporal data, such as the outputs of the hashing algorithm in a blockchain environment.

A new consensus mechanism proposed in [17] called Proof-of-Nonce was specifically designed to secure communications in smart mobility networks between vehicles. This mechanism was used to provide security in resource-limited and highly dynamic environments. This research is based on the idea of improving Nonce through a simple and effective model. The researcher was able to prove that Nonce analysis is important in enhancing blockchain security in light and distributed environments.

An improvement was made in RBF networks by combining the smart PSO algorithm with Levy Flight, which in turn led to a significant improvement in the prediction results when applied to nonlinear environmental systems in [18]. This study proved the actual success of Levy Flight techniques in improving models related to machine learning, which strengthens the hypothesis of combining them with the GRU neural network algorithm to evaluate the quality of nonlinear systems.

Deep reinforcement learning was used to design dynamic models in [19] to improve mining efficiency and verification in the blockchain-based edge network, focusing on increasing transaction efficiency and reducing latency. This study helped highlight how self-learning models can help adapt mining mechanisms to environmental changes, which may support the consideration of combining GWO and GRU as smart solutions.

The researchers in [20] proposed a hybrid evolutionary algorithm called Levy-flighted WSAR, which demonstrated superior efficiency in overcoming the local solution problem compared to standard algorithms. This researcher also supports the idea of combining Levy jumps with evolutionary algorithms.

In [21], researchers developed a hybrid framework for resource management in 5G networks using blockchain technology. The Levy Flight mechanism was combined with an algorithm inspired by the Energy Valley model to achieve a balance between security and data transfer speed, as well as improve resource efficiency. This study relied on a highly dynamic environment. As in 5G networks, this study used evolutionary algorithms to organize transactions and optimize resource allocation, emphasizing block verification within the blockchain environment. The results showed that the use of Levy Flight helped the proposed system avoid stagnation in local solutions while potentially achieving the greatest improvements in network performance.

In [22], researchers proposed an intelligent model using the PSO algorithm to improve the efficiency of blockchain in financial verification. The research results showed that relying on this type of intelligent algorithms inspired by nature contributes significantly to increasing computational efficiency and reducing mining time. The approach followed in this research was supported by these results, as the GWO algorithm was adopted and combined with the intelligent

neural model to predict the best value of the nons, which is an extension of the work that combines collective intelligence and deep learning in a lightweight blockchain environment.

The researcher combined evolutionary algorithms and deep learning algorithm to enhance the security of the Internet of Things network that is supported by blockchain in [23]. The researcher used the (GRU) model to classify the threats facing the questions and used the evolutionary algorithm (GOW) to improve the performance parameters, which in turn enhanced the ability of the proposed system to efficiently detect malicious nodes. The researcher explained how the GWO and GRU could work in harmony within complex and suspicious environments.

Researchers in [24] have proposed a new scientific proposal, a hybrid framework combining IEHO and ANFIS, to improve the efficiency of resource scheduling and the use of a blockchain-based robotic system for data transmission. The results of this research demonstrated the efficiency of hybrid systems in reducing energy consumption and adapting to the environment. These results support the research goal of improving a hybrid framework that combines Levy Flight and GWO with the GRU model to improve nonentity detection in a lightweight blockchain environment.

3. Gap of the research

Despite the progress and depth of research into the application of optimization algorithms in blockchain and AI environments, most previous studies have focused on security, resource scheduling, or overall performance, without focusing on the critical element of Proof of work— the process of discovering the value of Nonce. Many attempts have emerged to leverage nature-inspired algorithms, including collective intelligence algorithms like PSO and GWO. The Levy Flight mechanism has also proven effective in enhancing the algorithm's ability to avoid local solutions. Researchers have used sequential models like GRU to process temporal data, but they operate independently from optimization algorithms. Previous research shows that there has not been a study that combines GOW, which has been improved with Levy Flight, with the GRU model in a smart hybrid system that mixes deep learning and smart optimization to solve the mining problem more effectively while balancing prediction and exploration. This approach enhances verification speed and reduces energy consumption in lightweight blockchain environments, especially in environments with limited system resources.

5. Motivated of the research

Blockchain technologies have rapidly expanded into distributed and smart systems. Such growth has increased the need to improve mining mechanisms efficiently and lightly, especially in resource-constrained environments such as smart vehicles, wireless networks, and mobile devices. The process of discovering Nonce is one of the most complex and time-consuming in proof-of-work algorithms, requiring attention in the search for smart solutions that strike a balance between efficiency and performance. Optimization algorithms inspired by crowd intelligence, such as the Grey Wolf algorithm, have demonstrated promising potential in improving search mechanisms. Levy Flight has shown high efficiency and can expand the solution space, helping to avoid local solutions. We have also utilized smart networks like the GRU, which enable real-time data analysis and future outcome prediction. The main reason for starting this research was to combine these three technologies into one system that can speed up Nonce discovery and enhance mining operations in a lightweight blockchain setup, all while keeping security and performance intact. The main motivation was to provide a smart and practical solution that helps reduce time and energy consumption and increases system reliability in environments characterized by limited resources.

6. Methodology

The suggested system (Figure 1) uses a multi-level hybrid method that mixes statistical analysis, clustering algorithms improved by genetic algorithms (k-means), deep learning algorithms (GRU), and swarm intelligence algorithms (GOW) to create a smart and efficient way to find the best value of a random number (Nonce) in a lightweight blockchain setting. It consists of three basic stages that are interconnected in a way that enables each stage to enhance the effectiveness of the next.

1. Statistical Analysis and Intelligent Data clustering:

This is one of the important stages that begins with collecting data resulting from the hashing operation and then extracting a set of pivotal properties, such as:

- Nonce: random number.
- Scaled Nonce: It is a modified numerical representation of the original Nonce value, normalized or scaled to a specific range (such as [0, 1]) to facilitate its use in statistical models or neural networks.

It is calculated by subtracting the lowest Nonce value in the data set from the current Nonce value, then dividing the result by the difference between the highest and lowest Nonce values in the same set.

- Entropy: It is used to measure the expected extent of randomness or balanced distribution in the series. zero percent [25] defined as:

$$H(X) = -\Sigma\, p(xi) * log2(p(xi))$$

Where: H(X): a measure of randomness or uncertainty, p(xi) : The chance of event xi happening.

n: Data's total distinct character count.

- Diversity: Measures the frequency of values or symbols within the output.

Pattern Frequency [26] defined as:

$$diversity = 1 - \sum_{i=1}^{k} p_i^2$$

p_i: Frequency of the value or symbol i ,k = Total number of unique values or symbols in the output .

- Zero ratio: The ratio of the number of zeros at the beginning of each hash string to the number of bits being analyzed

$$zero\ ratio = \frac{Z}{D}$$

Where, Z: The number of zeros in the first digits of the hash, which is equal to the difficulty level.

D: Difficulty level.

- Frequency: In the context of hash function analysis, frequency refers to the number of times a bit (0) is repeated within the extracted portion of the hash function output. Frequency is used as an indicator of the internal distribution pattern of hash values. This frequency value is important in assessing the degree of homogeneity, which helps detect deviations or potential structural characteristics of the generation mechanism. Frequency (0–f) is calculated using the following formula:

$$F_0 = \frac{N_0}{L}$$

W here, F_0: The relative frequency of bit 0, N_O: The number of bits that hold the value (0)

L: The total length of the part of the fragmentation product that was looked at. If the expected (F_0) value is close to zero, for example (0.4) in the case of a random distribution, this indicates balanced segmentation behavior. However, if there is a noticeable deviation, this may indicate suboptimal properties that can be used for evaluation or improvement within the proposed system.

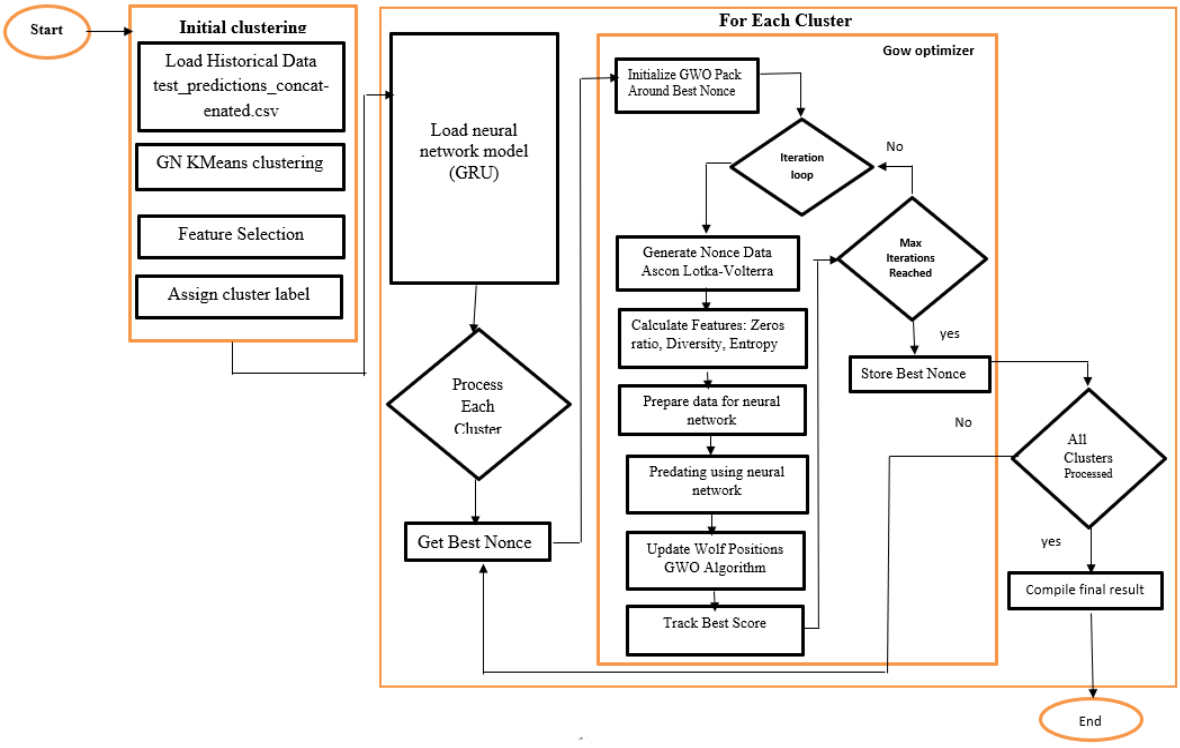

Figure 1. The architecture of the proposed system for Nonce Prediction in Blockchain environment.

Next, the proposed system groups the data using a method that takes advantage of both the fast K-Means algorithm for clustering and the flexible genetic algorithm for improving starting points and finding the best number of clusters (k). The goal of this intelligent clustering is to reduce dispersion within each cluster and achieve statistical homogeneity, which enhances the accuracy of optimization in subsequent stages. Each dataset (cluster) is handled independently based on a set of predefined statistical features, allowing the prediction and optimization stages to be customized according to the behavior of the data within each cluster. Algorithm (1) presents the pseudocode for the hybrid GA-KMeans algorithm, providing a detailed explanation of its working mechanism.

	Algorithm 1: GA-based KMeans Optimization
Input	• Dataset X • Range of k values: k range = (min k, max k). • Population size • Number of Generations: g • Mutation rate • Crossover rate
Output	- Best number of clusters (best k) - Scores for every k throughout generations (scores df)
Begin **END**	**Steps:** Start a random population of cluster numbers (k values) in side k range. From 1 to generations, for every generation, do: Determine the fitness of every member in the population. - For every k in the population: Run KMeans clustering using k clusters Compute the silhouette score. Designate the fitness value as the silhouette score. Choose the top half of the population depending on greatest fitness ratings. Do crossover: - - From the top people, randomly choose two parents. - Average the k values of the parents to produce a child - Include the child in the new population. l. Execute mutation: - For every person in the new population: - Change k by ±1 or ±2 with probability equal to mutation rate. - Keep changed k inside k range limits. e. Add the newly produced people to the population. At this generation, record the fitness ratings for every k value.

To ensure the optimal solution (k) for the number of clusters, a scientific comparison was conducted between the proposed algorithm (k-means GA) and another hybrid algorithm that also combines an intelligent optimization algorithm (stochastic gradient descent (SGD) algorithm) and the k-means algorithm(2) . This comparison was conducted with the aim of selecting the most efficient algorithm for data clustering according to the statistical properties extracted from the experimental values. The comparison was based on three metrics, which will be explained later.

	Algorithm 2: SGD-based KMeans Optimization
Input	D: Dataset • K min: Minimum number of clusters • K max: The most clusters number • iterations = number of optimization steps • learning rate = step size for updating k
Output	Best k: The best number of clusters Best score: The Silhouette Score that goes with it

Begin	Initialize: • Set best k ← random number between k min and k max, • Best score ← -∞ • Set current k to best k. For i = 1 to iterations, do: 1. Use current k clusters to run K-Means on the dataset D 2. Find the Silhouette Score: score → silhouette (D, labels of current k) 3. If score is greater than best score, best k ← current k best score ← score 4. Update current k: direction → a random choice of [+1, -1] New k ← current k + (direction × learning rate) Set new k to be between k min and k max. Then set current k ← new k.
END	Return best k and best score

7. predicting Nonce Quality Using a Neural network Model

At this point, a smart model that uses a gated recurrent neural network was used to predict how good the Nonce value is by looking at the data features, particularly the zero ratio. These zero ratios are an important measure for evaluating the quality of the Nonce within proof-of-work (POW) algorithms, as the strength of the encryption and the difficulty in discovering the optimal value of the Nonce depend on these zero ratios. The ability of the intelligent neural network (GRU algorithm3) model to learn from temporal and sequential data makes it very suitable for dealing with hash data. After completing the training of this model, the initial criterion for evaluating the quality of the Nonce was passed as input to the Gow algorithm to improve the Nonce value more accurately within each cluster.

	Algorithm3. GRU Model for zero predication
Input **Output**	Input Shape: Time Steps, Num Features compiled model for the expected proportion of zeros in the group A continuous predictive value that represents the expected zero ratio.
Begin	**Steps:** **1.** Create a Sequential Neural Network model **2.** Add the GRU Layer**.** - Units: 64 - Activation: ReLU - Input shape: Time Steps, Num Features (The purpose of this layer is to process sequential inputs and temporal learning)

End	**3.**Add Dense Layer: - Units: 32 - Activation: ReLU (Gives nonlinear relationships based on GRU outputs). **4.** Add a dense layer in output**.** **5.** - Units: 1 - Activation: Linear (generates an ongoing scalar output: expected Zero Ratio **6.** The model compile with: -Adams is the optimizer. -Loss function: MSE, mean squared error - Mean Absolute Error (MAE) as the evaluation criterion. 6. Return the created model.

Algorithm3 demonstrated a smart practical framework for building and training a neural network (GRU) model to predict the quality of Nonce values by predicting the percentage of zeros. This model begins by initializing the GRU layer, which consists of 64 units activated by the ReLU function. The dense layer, which consists of 32 layers, and finally the output layer, which is characterized by being single-layer with a linear activation function to generate a continuous output that reflects the expected percentage of zeros, follow this phase. The model is assembled using the Adam algorithm to improve the gradients, and to evaluate the accuracy of the model during training, the MSE and another metric was used.

This smart model added to the scientific knowledge of the proposed system, which may be summed up as follows:

1- It contributed to predicting the percentage of zeros for each set of data resulting from the hashing process.

2- It reduced the time required to find the best and most efficient solution for determining the initial values of high-quality Nonce.

3- This model contributed to improving the efficiency of the GOW algorithm, as the Wolf algorithm relies on the expected value of the Nonce to direct the search process toward spaces that are more likely to discover the ideal Nonce value, which reduces the number of attempts required and improves the speed of convergence.

4- It increased the accuracy of prediction and security in a lightweight blockchain environment by exploiting the unique statistical properties of each cluster.

5- In a lightweight blockchain environment, this model is of utmost importance in increasing the accuracy of prediction and security by exploiting the unique statistical properties of each cluster.

8. GWO-Levy (Optimization Phase)

This stage focuses on improving the Nonce (the random number used in the hash) found from the GRU model to identify the best value that meets the blockchain's difficulty requirements, using the Grey Wolf Optimizer (GWO) algorithm with Levy Flight-based improvements to better explore options and prevent being trapped in local solutions, as shown in Algorithm3.

	Algorithm4: A hybrid model of multithreaded GWO-Levy optimization with GRU-based evaluation
Input	• Initial Nonce: First guess for the Nonce value (output from GRU model) • data: data for hashing • difficulty: Number of zeros needed • max iterations: times to optimize • pack size: Number of wolves (possible Nonces)

	• max workers: Number of threads (not required).
Output	• final result: Data Frame of best Nonce statistics • iteration scores: List of fitness scores per iteration • best Nonce: Final optimized Nonce value
Begin	**Steps:** 1. Set up a shared hash cache and thread lock so that many threads can safely access it at the same time. 2. Define the function generate Nonce data cached (Nonce): a. If there is Nonce data in the cache → return result from cache b. Else: • Add Nonce to the data and find the Ascon Lotka–Volterra hash. • Find the Zero Ratio, Diversity, Entropy • Count how many times each character (0-9, a-f) appears • Keep everything in a Data Frame • Store the Data Frame in memory and return it. Frame 3. Set up the function prepare data: - Get features: Scaled Nonce, Diversity, Entropy, and Frequencies - Change the shape of the data to (1, num_features, 1) so it may be used as input for GRU. 4. Set up the function evaluate wolf(Nonce): • Use cached create Nonce data to make hash characteristics b. Prepare GRU input c. Use a trained GRU model to guess the Zero Ratio (fitness). d. Return (Nonce, expected score) 5. Set up the function evaluate pack parallel(wolves): - Use ThreadPool to check all of the wolves at the same time. - Give back a list of (Nonce, score) 6. Place wolves around the first Nonce with a small, random range. 7. For every iteration from 1 to max iterations a- Rate all the wolves at the same time and gain scores b- Choose wolves based on their fitness: - The best Nonce is alpha. - beta Nonce is the second-best - delta Nonce is the third best c- Calculate the adaptive parameter a = 2 - (iteration / max iterations)

	d - For every wolf in the pack: a. Use GWO formulae and the Levy step to find the new position: - A1, A2, A3 ← coefficients that change throughout time - C1, C2, and C3 are random coefficients. - D alpha, D beta, and D delta are the distances to the leaders. -X1, X2, and X3 are new positions. - Levy step = 1.5 * np. random. standard Cauchy() - new Nonce = the average of X1, X2, and X3 + the Levy step b. Make sure that Nonce is an integer that is not negative. **e-** Change the positions of the wolf pack to the new ones. 8. Generate final result: - Final result = generate Nonce data cached (data, difficulty, best Nonce). - return final result, iteration scores, best Nonce
END	

A hybrid algorithm4 based on the Grey Wolf (GWO) algorithm, supported by Levi Flight's multi-threaded random steps, is implemented to achieve efficient and fast random number optimization in the proposed blockchain system [26][27]. The process begins by initializing a certain number of wolves (random number filters) distributed around an initial value. A hash is generated for each random number using the ASCON Lotka–Volterra algorithm, developed as part of the Lightweight Blockchain System [28] after combining it with the underlying data. A set of statistical properties are then extracted from the resulting hash, including the zero percentage, diversity, entropy, and frequency of decimal symbols (F-0). These symbols are then reshaped to match the input format of a pretrained GRU neural network. We use this procedure to evaluate the quality of the random number by predicting the expected zero percentage, a crucial metric in Proof-of-Work systems. The GOW algorithm then sorts the wolves and selects the best ones (alpha, beta, and delta) based on the evaluation results. These then serve as a reference for updating the positions of the remaining wolves. The update process uses the Gow equations to find where each wolf should move next and then adds a random step called Levy diffusion to help explore new areas and prevent being stuck in one spot. Each iteration performs the update and evaluation of each wolf using multiple threads to ensure high and efficient performance. Several iterations repeat this process, culminating in the adoption of the best irrational value with the highest model evaluation. This system's use of GRU and clever exploration with the improved GWO-Levy algorithm could develop an effective way to find the best high-quality Nonce value in lightweight blockchain environments, especially in systems with limited resources [29][30].

9. Data set

The medical datasets are electronic medical records for a group of patients. The records have the patient's ID number, name, and date of birth, gender, medical condition, prescriptions, allergies, and the date of their last checkup. The dataset is only for testing and demonstration purposes. The URL for the dataset is http://filestore.nationalarchives.gov.uk/datasets/records/hospital-records.xls.

10. Extracted results and their analysis

In this section, the results obtained from the implementation of the proposed system are presented and analysed, and the performance of the model is evaluated based on a set of statistical criteria. First, to find the optimal number of clusters (k), we compared the proposed algorithm (k-means GA) with another hybrid algorithm that uses both an intelligent optimization technique (stochastic gradient descent (SGD)) and the k-means algorithm. This comparison was conducted with the goal of selecting the most efficient algorithm for the optimal value of the number of clusters, rather than choosing it randomly. The comparison was based on three metrics, as shown in Table 1 and Figure 2.

Table 1: A comparison of clustering metrics between the SGD–KMeans and GA–KMeans methods.

Metric	K-mean	GA– K mean Algorithm	SGD-K means Algorithm
Silhouette Score	0.8434	0.9980	0.9947
Davies-Bouldin Index	0.4605844	0.0000	0.0798
Calinski-Harabasz index	0.000	2.8824	0.000

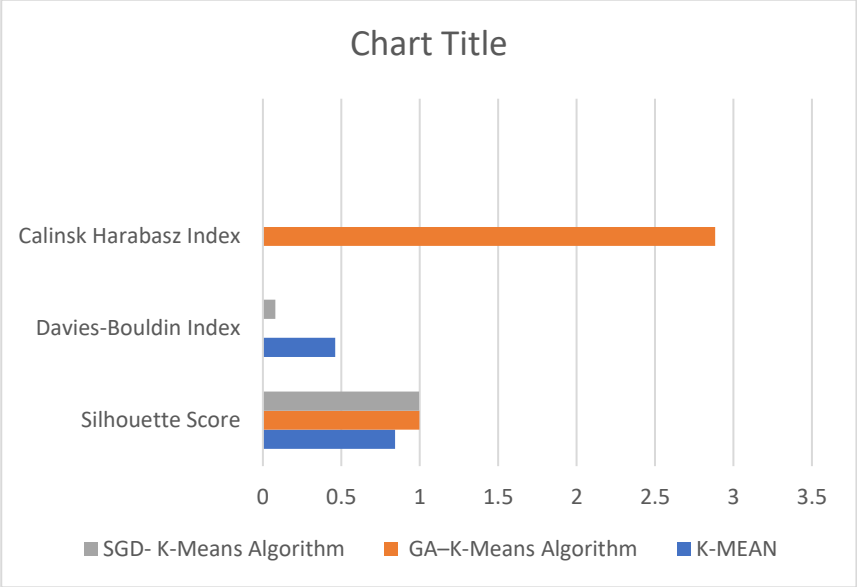

Figure 2. Comparative analysis of clustering metrics between SGD–KMeans and GA–KMeans algorithms.

The results shown in Table1 and Figure2 clearly demonstrate the superiority of the proposed algorithm, GA-KMeans, over both SGD-KMeans and K-Means on all evaluation metrics, as follows:

- GA-KMeans achieved the highest silhouette score value of 0.9980, reflecting near-perfect homogeneity within the cluster, compared to SGD-KMeans, which achieved 0.9947 and only 0.8434 for the traditional K-Means.

Table 2: Model performance metric comparison for (4) difficulty base on medical data set

Model	MSE	RMSE	MAE	MAPE	R^2 Score	Explained Variance	Mean Residual	Std Residual	Max Error	Min Error	Median Error
LSTM	0.0000	0.0000	0.0000	0.0001	1.0000	1.0000	0.0000	0.0000	0.0005	0.0000	0.0000
GRU	0.0000	0.0000	0.0000	0.0000	1.0000	1.0000	0.0000	0.0000	0.0002	0.0000	0.0000
CNN2D	0.0000	0.0000	0.0000	0.0000	1.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
LSTM + CNN	0.0000	0.0001	0.0000	0.0001	1.0000	1.0000	-0.0000	0.0000	0.0006	0.0000	0.0000
GRU + CNN	0.0000	0.0001	0.0001	0.0001	1.0000	1.0000	-0.0001	0.0000	0.0003	0.0000	0.0001
LSTM + GRU	0.0000	0.0001	0.0001	0.0001	1.0000	1.0000	0.0000	0.0001	0.0008	0.0000	0.0000

Looking at the results in Table2, which show how well six deep models predicted zeros from the hash function using a medical dataset at difficulty level 4, we can conclude the following:

The CNN2D model achieved optimal numerical performance, with all error metrics (MSE, RMSE, MAE, and MAPE) reaching zero with $R^2 = 1.0000$. However, these results indicate a high degree of over optimization, as reaching a maximum error of zero is unrealistic, suggesting that this model may have overfitted the data at the expense of its ability to generalize to other datasets. In contrast, the GRU model performed close to optimally, with a maximum error of 0.0002, the lowest among all models, and showed no clear signs of overlearning. The LSTM model performed similarly to the GRU model, with very low error metrics, but its maximum error was 0.0005, which is relatively high, making it less stable. The combined LSTM + CNN model achieved a high degree of accuracy, but its maximum error was 0.0006, indicating a slight decrease in generalization ability compared to the GRU and LSTM models. The hybrid GRU + CNN model achieved a maximum error of 0.0003, reflecting its good generalization ability and high accuracy, It is considered a more robust and accurate model than the GRU model. The LSTM + GRU model did not do well on various error measurements, with a maximum error of 0.0008, showing it has more inconsistency in learning too much and is therefore less dependable than the GRU and GRU + CNN models

Table 3: Model performance metric comparison for (6) difficulty base on health care data set

Model	MSE	RMSE	MAE	MAPE	R^2 Score	Explained Variance	Mean Residual	Std Residual	Max Error	Min Error	Median Error
LSTM	0.0000	0.0001	0.0000	0.0000	1.0000	1.0000	-0.0000	0.0001	0.0127	0.0000	0.0000
GRU	0.0000	0.0001	0.0000	0.0000	1.0000	1.0000	-0.0000	0.0001	0.0812	0.0000	0.0000
CNN2D	0.0000	0.0001	0.0000	0.0001	1.0000	1.0000	0.0000	0.0001	0.0701	0.0000	0.0000
LSTM + CNN	0.0000	0.0002	0.0001	0.0002	1.0000	1.0000	-0.0001	0.0002	0.0024	0.0000	0.0000
GRU + CNN	0.0000	0.0001	0.0000	0.0001	1.0000	1.0000	-0.0000	0.0001	0.0084	0.0000	0.0000
LSTM + GRU	0.0000	0.0000	0.0000	0.0001	1.0000	1.0000	0.0000	0.0000	0.0030	0.0000	0.0000

Table3 presents the results from the medical data set, based on difficulty level 6.The GRU model achieved the highest numerical accuracy among all models. However, its high maximum error (Max Error = 0.0812) indicates signs of overtraining, which reduces its ability to generalize when dealing with other data. The CNN2D model also achieved ideal results, but these may be unrealistic in real-world application environments, which also indicates overfitting. The combined LSTM + GRU model demonstrated good performance in terms of balance and reliability, combining high prediction accuracy with low error dispersion (Max Error = 0.0030), with very slight signs of overfitting. This makes it the ideal choice for practical applications in environments that rely on predicting the percentage of zeros resulting from a hash function

Accordingly, the LSTM + GRU model can be considered the best choice due to its robustness, predictive stability, and high generalization ability across difficulty level=6.

Table 4: Model performance metric comparison for (8) difficulty base on health care data set

Mode	MSE	RMSE	MAE	MAPE	R^2 Score	Explained Variance	Mean Residual	Std Residual	Max Error	Min Error	Median Error
LSTM	0.0000	0.0005	0.0004	0.0010	0.9998	0.9999	-0.0004	0.0003	0.0219	0.0000	0.0003
GRU	0.0000	0.0001	0.0000	0.0000	1.0000	1.0000	-0.0000	0.0001	0.0136	0.0000	0.0000
CNN2D	0.0000	0.0002	0.0001	0.0002	1.0000	1.0000	-0.0001	0.0001	0.0259	0.0000	0.0001
LSTM + CNN	0.0000	0.0011	0.0008	0.0021	0.9991	0.9996	0.0008	0.0007	0.0066	0.0000	0.0005
GRU + CNN	0.0000	0.0002	0.0001	0.0004	1.0000	1.0000	-0.0001	0.0001	0.0138	0.0000	0.0001
LSTM + GRU	0.0000	0.0004	0.0004	0.0010	0.9999	1.0000	0.0004	0.0002	0.0208	0.0000	0.0003

Finally, the results are shown in table4, which are based on difficulty level 8 and were conducted on the medical data set:

The GRU model did much better than all the other prediction models, showing the lowest error scores (MSE, RMSE, MAE, and MAPE) and the highest R^2 value (= 1.0000), which means it is very accurate and stable. This model also had the smallest maximum error (Max Error = 0.0136), showing it does not make extreme predictions very often. While the CNN2D model performed well overall, its higher maximum error (Max Error = 0.0259) suggests it may have focused too much on the training data, making it less effective with new data. This model also recorded the lowest error (Max Error = 0.0136), indicating limited dispersion in extreme predictions. Although the CNN2D model achieved excellent and optimal performance in its outputs, the high maximum error (Max Error = 0.0259) indicates overfitting, meaning the model may have learned too much of the data at the expense of its ability to generalize to other data. The hybrid LSTM + CNN model achieved the highest RMSE and MAPE values, as well as the highest average error (Mean Residual = 0.0008). This value indicates the model's weak prediction and stability, making it the least reliable. We found the models to be comparable. Combined models like LSTM + GRU showed decent performance, but they had higher maximum error values than the GRU model, which suggests that their predictions varied more and were less accurate in some situations, even though they still performed reasonably well overall. While hybrid LSTM + GRU models demonstrated acceptable performance, which was relatively comparable to that of the GRU model, they recorded higher maximum error values. This indicates variation in predictions and a loss of accuracy in some cases, despite maintaining a reasonable level of performance indicators.

Based on the above, it is recommended to use the GRU model as a reliable one at a difficulty score of 8, due to its effective balance between statistical accuracy and generalization ability.

Given the pivotal role difficulty plays in blockchains, especially in Proof-of-Work (PoW) systems, choosing an appropriate difficulty level is a crucial factor in determining the quality of predicting the percentage of zeros generated by a hash function. The conducted a comparison between three different difficulty levels: 4, 6, and 8. This comparison aims to determine which of these levels provides the most accurate and stable performance within the proposed lightweight blockchain environment, thus enhancing prediction efficiency and quality.

Table 5: comparison between three difficulty level (4,6,8)

Model	Performance of Difficulty (4)	Performance of Difficulty (6)	Performance Difficulty of (8)
GRU	Excellent performance, low maximum error (0.0002), no overfitting	good performance, but relatively high maximum error (0.0812), overfitting	the best overall, very high accuracy and low maximum error (0.0002), and remarkable stability
LSTM	Excellent performance, maximum error = 0.0005	Good performance, maximum error = 0.0127	good performance, maximum error = 0.0004
CNN2D	Apparently perfect performance, all values = 0, Overfitting is evident	Unreliable performance, maximum error = 0.0701, Overfitting is certain	Unrealistic performance, all errors = 0, overfitting assured
GRU + CNN	Very good performance, maximum error = 0.0003	Good performance, maximum error = 0.0084	good performance, maximum error = 0.0003
LSTM + CNN	Good performance, maximum error = 0.0006	Good performance, maximum error = 0.0024	Very good performance, maximum error = 0.0005
LSTM + GRU	Fluctuating performance, maximum error = 0.0008	Best within this grade, maximum error = 0.0030	Good performance, maximum error = 0.0004

The results of a comparative analysis of difficulty levels (4, 6, 8) in table5 showed that the most suitable difficulty level for blockchain applications is 8. AI models, especially the GRU model, achieved high accuracy performance with minimal error and stability, with no signs of overfitting. To provide high security within a blockchain environment, increasing the difficulty level makes the hash function output more difficult, which contributes to enhancing the system's resistance to rehashing or pattern prediction attacks. These are essential factors for enhancing the security and reliability of Proof-of-Work (POW) systems. This level also ensures better compatibility with the lightweight characteristics of blockchain, which must strike a delicate balance between computational efficiency and security. Therefore, difficulty level (8) was adopted in the proposed system design to achieve the highest levels of performance and security.

11. Conclusion and future work

This research presents an intelligent, integrated framework aimed at improving the Nonce discovery mechanism in lightweight blockchain environments, especially those that rely on proof-of-work mechanisms. This is achieved by integrating intelligent predictive models based on deep learning with evolutionary clustering and intelligent optimization algorithms. The study focused primarily on replacing energy- and resource-intensive random search mechanisms with an intelligent predictive approach based on learning from data. Through extensive experiments on health data, several neural models (GRU, LSTM, CNN2D), as well as hybrid models between them, were tested across multiple difficulty levels (4, 6, and 8). Based on these results, we note that the GRU model performed exceptionally well in terms of reliability and accuracy, outperforming other models on all prediction metrics and error indicators. This makes it the most suitable choice in resource-constrained blockchain environments. On the other hand, the smart hybrid algorithm GA–KMeans has proven its role in improving data before entering the prediction phase, which contributed to increasing the speed of reaching the best Nonce section in the search space. After comparing the hybrid models, it was found that some of them provided acceptable results, but with a noticeable difference in the maximum error, which reduces their reliability in real blockchain environments. Based on this, it can be said that the framework proposed in this research paper represents a scientific and practical contribution to smart blockchain applications and provides a smart alternative to traditional proof-of-work mechanisms, especially in applications that require a balance between predictive accuracy and computational efficiency, such as healthcare and tax systems.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf.

[2] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Proc. 2017 IEEE Int. Congr. Big Data (BigData Congr.)*, 2017, pp. 557-564.

[3] R. H. Salman and H. B. A. Wahab, "Enhancing blockchain security and decentralization using the ASCON algorithm for lightweight applications," *IEEE Access*, vol. 12, pp. 48570-48583, 2024.

[4] J. Yli-Huumo, D. Ko, S. Choi, S. Park, and K. Smolander, "Where is current research on blockchain technology?—a systematic review," *PLOS ONE*, vol. 11, no. 10, p. e0163477, Oct. 2016.

[5] M. A. Alzubi, R. H. Salman, and H. B. A. Wahab, "Using lightweight post-quantum algorithms for enhancing blockchain security," *J. Inf. Process. Syst.*, vol. 19, no. 1, pp. 1-15, Feb. 2024.

[6] W. Wang *et al.*, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22328-22370, 2019.

[7] M. Swan, *Blockchain: Blueprint for a New Economy*. Sebastopol, CA, USA: O'Reilly Media, 2015.

[8] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292-2303, 2016.

[9] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On blockchain and its integration with IoT: Challenges and opportunities," *Future Gener. Comput. Syst.*, vol. 88, pp. 173-190, Nov. 2018.

[10] A. de Vries, "Bitcoin's growing energy problem," *Joule*, vol. 2, no. 5, pp. 801-805, May 2018.

[11] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *Proc. Int. Conf. Financ. Cryptogr. Data Secur.*, 2014, pp. 436-454.

[12] G. T. Nguyen and K. Kim, "A survey about consensus algorithms used in blockchain," *J. Inf. Process. Syst.*, vol. 14, no. 1, pp. 101-128, Feb. 2018.

[13] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Gener. Comput. Syst.*, vol. 107, pp. 841-853, Jun. 2020.

[14] M. J. Dworkin, "SHA-3 standard: Permutation-based hash and extendable-output functions," Nat. Inst. Stand. Technol., Gaithersburg, MD, USA, NIST FIPS 202, Aug. 2015.

[15] [15] M. Conti, E. S. Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of bitcoin," *IEEE Commun. Surv. Tutor*, vol. 20, no. 4, pp. 3416-3452, 4th Quart. 2018.

[16] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46-61, Mar. 2014.

[17] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*, 2nd ed. Luniver Press, 2010.

[18] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv:1412.3555*, Dec. 2014.

[19] N. Y. Ahn and D. H. Lee, "Secure vehicle communications using proof-of-nonce blockchain," *arXiv: 2011.07846*, Nov. 2020.

[20] R. Li and W. Ying, "Improved particle swarm optimization based on Lévy flights," *J. Syst. Simul.*, vol. 29, no. 8, pp. 1685-1692, 2020.

[21] D. C. Nguyen *et al.*, "Intelligent blockchain-based edge computing via deep reinforcement learning: Solutions and challenges," *IEEE Netw.*, vol. 36, no. 6, pp. 12-19, Nov./Dec. 2022.

[22] A. Baykasoğlu and M. E. Şenol, "WSAR with Levy flight for constrained optimization," in *Proc. 7th Int. Conf. Harmony Search, Soft Comput. Appl. (ICHSA 2022)*, 2022, pp. 217-225.

[23] K. S. Kumar *et al.*, "A secure and efficient blockchain and DLT-based optimal resource management in digital twin beyond 5G networks using hybrid energy valley and Lévy flight distributor optimization algorithm," *IEEE Access*, vol. 12, pp. 39902-39921, 2024.

[24] N. A. Saqib and S. T. AL-Talla, "Scaling up security and efficiency in financial transactions and blockchain systems," *J. Sens. Actuator Netw.*, vol. 12, no. 2, p. 31, Apr. 2023.

[25] F. M. Alserhani, "Integrating deep learning and metaheuristics algorithms for blockchain-based reassurance data management in the detection of malicious IoT nodes," *Peer-to-Peer Netw. Appl.*, vol. 17, no. 6, pp. 3856-3882, Nov. 2024.

[26] P. A. Raj, M. Rajakumaran, S. P. Murugan, and S. Senthilkumar, "Hybridization of metaheuristic algorithms for resource scheduling in distributed robotic control system," *Discov. Appl. Sci.*, vol. 7, no. 5, p. 210, May 2025.

[27] V. M. Eskov, V. V. Eskov, Y. V. Vochmina, D. V. Gorbunov, and L. K. Ilyashenko, "Shannon entropy in the research on stationary regimes and the evolution of complexity," *Mosc. Univ. Phys. Bull.*, vol. 72, no. 3, pp. 309-317, 2017.

[28] J. E. McLaughlin, G. W. McLaughlin, J. S. McLaughlin, and C. Y. White, "Using Simpson's diversity index to examine multidimensional models of diversity in health professions education," *Int. J. Med. Educ.*, vol. 7, pp. 1-6, Jan. 2016.

[29] O. A. Hassen *et al.*, "Towards a secure signature scheme based on multimodal biometric technology: Application for IoT blockchain network," *Symmetry*, vol. 12, no. 10, p. 1699, Oct. 2020.

[30] A. H. Alzubi, R. H. Salman, and H. B. A. Wahab, "Enhancing blockchain security using lightweight post-quantum algorithms," *J. Inf. Process. Syst.*, vol. 19, no. 1, pp. 1-15, Feb. 2024.