# INTELLIGENT DATA GOVERNANCE ALGORITHM OPTIMIZATION AND SYSTEM DESIGN BASED ON DEEP LEARNING

Qiang LING[1*]

*As financial data continues to expand at a rapid pace and security standards become more stringent, intelligent data governance encounters hurdles like inadequate accuracy in time - series predictions, slow encryption processes, and limited resistance to attacks. This research endeavors to create an intelligent data governance framework that combines effective prediction capabilities with robust security safeguards by merging deep - learning and cryptographic techniques. The research approaches are as follows: First, an enhanced dual - path Hopfield neural network (DPHN) is introduced. This network integrates a hybrid approach combining δ learning rules and Hebb rules to fine - tune the weight - updating process, thereby enhancing the accuracy of predicting the closing price of the CSI 300 Index over time. Second, a configurable round transformation (CRT - AES) version of the AES encryption algorithm is devised. It incorporates dynamic key generation and a randomized sequence of round transformations to strike a balance between security and computational demands. Finally, a mechanism to counteract power analysis attacks is established by integrating the Hamming distance model and random mask technology. The experimental findings reveal that the DPHN model achieves a root mean square error (RMSE) of 18.32 on the test set, a 25.9% reduction compared to the traditional Hopfield model. It also boasts a coefficient of determination ($R^2$) of 0.972. For the CRT - AES encryption, the throughput reaches 5.21Gbps, a 38.1% increase over standard AES - 128. Moreover, the success rate of differential fault attacks is minimized to $10^{-6}$ through dynamic round transformations. Security analysis indicates that the system's key entropy stays at 4.78 bits even after 16 fault injections, with a false - positive rate below 5%. This study offers a fresh paradigm for data lifecycle governance in highly sensitive areas such as finance. By integrating disciplines at the algorithm level, it overcomes the limitations of traditional methods in dynamic feature modeling, real - time encryption, and side - channel defense, which holds significant practical value for advancing the engineering application of intelligent data governance technology.*

**Keywords:** Hopfield Neural Network, AES Encryption Optimization, Power Analysis Attack, Time Series Prediction, Deep Learning

## 1. Introduction

As the volume of data in the financial sector surges exponentially and security threats become increasingly intricate, intelligent data governance technology confronts numerous obstacles. These include inadequate accuracy in time - series

[1] * Information Center, China Coal Technology and Engineering Group Corporation, Beijing, China, e-mail: lingqiang@ccteg.cn

predictions, sluggish encryption processes, and limited resistance to attacks [1 - 2]. In the present research, the traditional Hopfield neural network exhibits issues like sluggish convergence rates and getting trapped in local optima when predicting financial time - series data. Meanwhile, the standard Advanced Encryption Standard (AES) encryption algorithm struggles to strike a balance between real - time performance and the need to resist side - channel attacks. Current methods have failed to effectively address the challenge of jointly optimizing security protection and computational efficiency in dynamic data environments [3 - 5].

This paper centers on the cross - disciplinary innovation of deep learning and cryptographic technologies. It proposes an enhanced Hopfield network model featuring a dual - path hybrid learning mechanism. This mechanism enhances the model's ability to capture time - series features through the dynamic blending of δ rules and Hebb rules. Additionally, an AES encryption algorithm with a configurable round transformation architecture is designed. This enables dynamic key generation and the randomization of the order in which round functions are executed. A multi - tiered defense system is also constructed, based on the Hamming distance model and random mask technology. The research has been validated through predictions of the CSI 300 index and NIST standard tests. It significantly boosts prediction accuracy and encryption efficiency, while also strengthening the system's ability to withstand differential fault attacks. Thus, it offers a comprehensive solution for interdisciplinary integration in data governance within highly sensitive fields.

## 2. Hopfield Neural Network Analysis

### *2.1. Hopfield Neural Network*

The Hopfield neural network is a single-layer neural network with feedback connection weight for terminological accuracy in neural network theory. In the same paragraph, indices are not well-formatted, the output xi is multiplied by the connection weight and acts on the jet neuron [6-7]. Each neuron receives input from other neurons, and then outputs it after being processed by the activation function, which is generally used $f_1$ $f_2$ ,. $f_n$ denotes its state-activated function, and $i_f$ denotes its threshold function [8]. DHNN generally chooses the same activation function, and the expression is as follows:

$$f_1(x) = f_2(x) = \ldots = f_n(x) = \operatorname{sgn}(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases} \tag{1}$$

In the above formula, $\operatorname{sgn}(x)$ is a symbolic function.

There are generally two ways in which the network works: asynchronous and synchronous. Among them, the asynchronous working mode is the most widely used, and the so-called asynchronous working mode is that the value of only one neuron changes at a time when the network is running $i$, and the value of other

neurons remains the same, and the state can be calculated according to the following formula [9-11]:

$$x_j(t+1)=\begin{cases}\mathrm{sgn}\left[\ \mathrm{net}\ _j(t)\right] & j=i\\ x_j(t) & j\neq i\end{cases} \tag{2}$$

In the above formula, $x_j$ Network output.

Neurons undergoing changes can be chosen in a pre-set sequence or picked randomly. Each time a neuron modifies its state, it assesses whether to change based on the net positive and negative values of its inputs. This ensures that the state of neurons doesn't shift every single time [12]. The second approach is the synchronous mode of operation. In this mode, whenever the network experiences a data alteration, all the neurons within the entire network change their states simultaneously:

$$x_j(t+1)=\mathrm{sgn}\left[\mathrm{net}_j(t)\right]\quad j=1,2,\cdots,n \tag{3}$$

Artificial neuron network learning algorithms are generally divided into three methods: δ learning rules, Hebb learning rules, and competitive learning rules.

(1) δ Learning rules, also known as corrective learning rules. When the actual output of neuronal I know k, then the magnitude of its error is:

$$x_j(t+1)=\mathrm{sgn}\left[\mathrm{net}_j(t)\right]\quad j=1,2,\cdots,n \tag{4}$$

Then the process of network training can be regarded as the process of finding the minimum, so corrected error learning has become a typical optimization problem. At the same time, the commonly used objective function is to find the minimum value of the mean square error, i.e.:

$$Y=E\left\{\frac{1}{2}\sum_{i=1}^{N}(t_i-y_i)^2\right\} \tag{5}$$

where E is the statistical expectation operator and Y is the objective function. Since the process of the DHNN network in finding attractors is the process of continuous updating and changing of the weight matrix, the above problem can be transformed into $w_{ij}$ the minimum value of the weight of Y, and then the correction of the weight can be obtained by using the steepest gradient descent method, i.e.:

$$\Delta w_{ij}(k)=\eta\cdot E_i(k)\cdot f^{'}(w_i x)\cdot x_j(k) \tag{6}$$

where, η is the learning rate, $f^{'}$ is an activation function.

(2) Hebb

The Hebb learning rule, that is, when the neuronal state at both ends of the connection is synchronized, then the strength of the connection increases, and vice versa, it decreases, which is mathematically expressed as [13-15].:

$$\Delta w_j(k)=F\left(y_i(k),x_i(k)\right) \tag{7}$$

where, $y_i(k),x_i(k)$ respectively $w_g$ Neuronal states at both ends, one of which is

commonly used is:

$$\Delta w_i(k) = \eta \cdot y_i(k) \cdot x_i(k) \tag{8}$$

In the following formula, the key variables are defined as follows:

$f'$ : the derivative of the neuronal activation function.

$E(k)$ : the prediction error of the kth iteration.

$y(k)$ : The network output value of the kth iteration.

Equation (6) is based on $\delta$ rule weight amendments, where: $\eta$ for the learning rate; Equation (7)-(8) is the expression for updating the weight under the Hebb rule, $x_i$ and $x_j$ Neuronal state, respectively.

### *2.2. Model comparison and verification*

The CSI 300 Index tracks 300 large-cap, liquid stocks representative of the Shanghai and Shenzhen exchanges, serving as a key barometer for the overall A-share market. This study forecasts the index's short-term closing price using data from January 4, 2011, to December 30, 2022. Given the limited short-term impact of fundamental data on the index, ten daily trading features are selected as predictors: opening price, closing price, high price, low price, trading volume, turnover, price amplitude, price change, price change rate, and turnover rate [16-17].

Inputs to the forecasting model consist of rolling 20-day windows of these ten features. The target label for each window is the closing price on the subsequent trading day. The dataset is partitioned chronologically based on label dates: samples predating July 1, 2021, are split 4:1 into training and validation sets for model development, while samples from July 1, 2021, onward form the test set for prediction evaluation. All data originates from the Choice Financial Terminal database.

The CP - Hopfield model presented in this context is designed to tackle a regression issue, specifically forecasting the exact closing price. To gauge its performance in comparison to benchmark models, prediction accuracy is measured through several metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and the Coefficient of Determination ($R^2$). The corresponding formulas for these metrics are outlined as follows:

The CP - Hopfield model presented in this context is designed to tackle a regression issue, specifically forecasting the exact closing price. To gauge its performance in comparison to benchmark models, prediction accuracy is measured

through several metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and the Coefficient of Determination ($R^2$). The corresponding formulas for these metrics are outlined as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{9}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{10}$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - y_i}{y_i}\right| \tag{11}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(\bar{y}_i - y_i)^2} \tag{12}$$

In comparison to Recurrent Neural Networks (RNNs), Gated Recurrent Units (GRUs) of a similar nature, and non - recurrent Multi - Layer Perceptron (MLP) models, the Hopfield model exhibits greater prowess in handling time - series data and extracting feature details more efficiently [18]. To validate this assertion, we developed RNN and GRU models that share the same structural parameters as the Hopfield model. Moreover, an MLP model with hidden layers comprising 128, 64, and 32 neurons respectively was also crafted. These four foundational models underwent training on the same input samples and were subsequently utilized to generate predictions using test set data.

The prediction outcomes of the basic models are presented in Fig. 1.

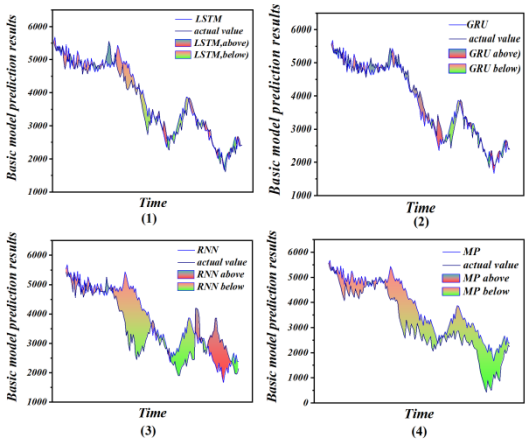


Fig. 1. Comparison of actual and forecasted results

As depicted in Fig. 1, Hopfield, GRU, and RNN models, which are capable of processing time - series data, provide closing price predictions that are closer to the actual closing price. In contrast, the MLP model shows the largest deviation from the actual closing price, and this deviation persists over the longest time interval.

For a forecasting model, the extent to which its predictions align with the actual closing price of the index is crucial in determining its practical utility in investment scenarios. A model that fits the actual data better offers more accurate forecasts, leading to lower risks when applying its predictions in investment practices and thus having greater practical value.

## 3. Intelligent data governance algorithm based on deep learning

### 3.1. Information confidentiality algorithms

The AES cipher algorithm is a block-symmetric cipher algorithm, which has characteristics of security, wide application field, and convenient implementation. The length of the input plaintext is 128 bits, and the length of the input key can be 128 bits, 192 bits, and 256 bits. This is shown in Table 1.

*Table 1.*

**AES algorithm classification**

| AES | Key length (32bits) | Packet length (32bits) | Number of crypto rounds |
|---|---|---|---|
| AES-128 | 4 | 4 | 10 |
| AES-192 | 6 | 4 | 12 |
| AES-256 | 8 | 4 | 14 |

The encryption formula for the AES algorithm is:

$$C=E(K, P) \tag{13}$$

Taking AES-128 as an example, the input 128-bit plaintext and 128-bit keys are first divided into 16 bytes, which are defined as:

$$P = P_0, P_1, \cdots, P_n \tag{14}$$

$$K = K_0, K_1, \cdots, K_n \tag{15}$$

The core innovation of CRT-AES lies in the dynamic rotation architecture (as shown in Figure 2):

Configurable Round Sequence: Execute the order by randomizing the round function (e.g.).Sub Bytes→Shift Rows→Mix Columns→ AddRoundKeyorShiftRows→Sub Bytes→Arounder→Mix Columns),breaking the fixed round pattern;

Dynamic key expansion: Each round of key generation introduces the previous ciphertext hash as a perturbation factor to enhance key correlation [19]

All four of the transformations are reversible. Below, we will delve into these transformations and their corresponding inverse operations.

(1) Row shift transformations.

The row shift and retrograde shift transformations are cyclic shifts to the data. The specific process of row displacement is shown in Fig. 2:

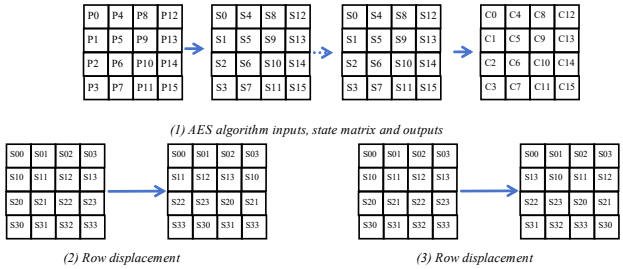


Fig. 2. Row shift and retrograde shift transformations.
(State matrix annotation standard: S it represents the j column bytes of row I (0⩽I, j⩽3);
Shift direction: the nth row loop shifts n bytes to the left (n=0,1,2,3).
The red arrow indicates the movement of the byte. ")

It's evident that the outcome of the third line - shifting operation is identical to that of the retrograde shift. Consequently, you can employ simplified code to cut down on the utilization of selectors during the code - design phase.

(2) Column blending and reverse column blending

Column hybrid transformations are implemented by matrix operations based on finite fields. The state matrix is left multiplied by a fixed matrix and calculated on the finite field GF (15) to obtain the confused state matrix, as shown in the public (16).:

$$\begin{bmatrix} S'_{0,0} & S_{0,1} & S_{0,2} & S'_{0,3} \\ S'_{1,0} & S'_{1,1} & S'_{1,2} & S'_{1,3} \\ S'_{2,0} & S_{2,1} & S'_{2,2} & S'_{2,3} \\ S'_{3,0} & S'_{3,1} & S_{3,2} & S'_{3,3} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix} \tag{16}$$

The transformation operation in column j, where the transformation operation can be represented by equation (17):

$$S_{0,j} = (2 \cdot S_{0,j}) \oplus (3 \cdot S_{1,j}) \oplus S_{2,j} \oplus S_{3,j} \tag{17}$$

$$S_{1,j} = S_{0,j} \oplus (2 \cdot S_{1,j}) \oplus (3 \cdot S_{2,j}) \oplus S_{3,j} \tag{18}$$

$$S_{2,j} = S_{0,j} \oplus S_{1,j} \oplus (2 \cdot S_{2,j}) \oplus (3 \cdot S_{3,j}) \tag{19}$$

$$S_{3,j} = (3 \cdot S_{0,j}) \oplus S_{1,j} \oplus S_{2,j} \oplus (2 \cdot S_{3,j}) \tag{20}$$

For the finite GF (18) multiplication and addition calculations in the column mixed transformation matrix, please refer to the calculation methods introduced in Chapter 2. For example, the state matrix of the input is as follows:

$$\begin{bmatrix} C9 & E5 & FD & 2B \\ 7A & F2 & 78 & 6E \\ 63 & 9C & 26 & 67 \\ B0 & A7 & 82 & E5 \end{bmatrix} \tag{21}$$

Take the first column of operations as an example:

$$S'_{0,0} = (2 \cdot C9) \oplus (3 \cdot 7A) \oplus 63 \oplus B0 = D4 \tag{22}$$

$$S'_{1,0} = C9 \oplus (2 \cdot 7A) \oplus (3 \cdot 63) \oplus B0 = 28 \tag{23}$$

$$S'_{2,0} = C9 \oplus 7A \oplus (2 \cdot 63) \oplus (3 \cdot B0) = BE \tag{24}$$

$$S'_{3,0} = (3 \cdot C9) \oplus 7A \oplus 63 \oplus (2 \cdot B0) = 22 \tag{25}$$

The inverse column hybrid transformation serves as the reverse operation of the column mixture [20-21]. The left - multiplication matrix used in the column mixture and the one for the inverse column hybrid transformation are inverse matrices of each other. The formula for the inverse column hybrid transformation is presented in Equation 26:

$$\begin{bmatrix} S'_{0,0} & S'_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S'_{1,2} & S_{1,3} \\ S'_{2,0} & S'_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix} = \begin{bmatrix} 05 & 00 & 04 & 00 \\ 00 & 05 & 00 & 04 \\ 04 & 00 & 05 & 00 \\ 00 & 04 & 00 & 05 \end{bmatrix} \left( \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix} \right) \tag{26}$$

### 3.2. Security analysis of symmetric algorithms

The typical procedure for a differential fault attack targeting a symmetric cryptographic algorithm unfolds as follows: Initially, the symmetric cryptographic algorithm is employed to process a randomly generated message, allowing the attacker to acquire the correct output. Subsequently, the symmetric cipher algorithm is executed once more to handle the identical plaintext. During this second processing run, a random single - word fault is injected, and the resulting faulty output is captured. Finally, leveraging the pairs of correct and faulty outputs that have been gathered, along with relevant theorems and differential analysis techniques, the subgrouping information of the messages used in the last round is retrieved. This process is repeated until all subgroups are recovered. Following that, the currently utilized output message is reconstructed based on the message processing scheme.

According to the processing process of the symmetric cryptographic algorithm, only the corresponding processing is carried out for module B in each step, and B64 is:

$$B_{64} = \left( A_{63} + f_{63}\left( B_{63}, C_{63}, D_{63} \right) + W[R(63)] + T_{63} \right) \square \ s[63] + B_{63} \tag{27}$$

For the symmetric cryptographic algorithm, the output is Y.

$$\mathrm{Y} = \left( \mathrm{Y}_0, \mathrm{Y}_1, \mathrm{Y}_2, \mathrm{Y}_3 \right) = \left( A_{64} + A_6, B_{64} + B_0, C_{64} + C_0, D_{64} + D_0 \right) \tag{28}$$

Bring in a known initial value to be available.

$$\begin{aligned} \mathbf{B}_{63} &= \mathbf{C}_{64}, \\ \mathbf{C}_{63} &= \mathbf{D}_{64}, \\ \mathbf{D}_{63} &= \mathbf{A}_{64}, \\ \mathbf{R(63)} &= \mathbf{9}, \\ \mathbf{s[63]} &= \mathbf{21}, \end{aligned} \tag{29}$$

$$B_{64} = \left( A_{63} + f_{63}\left( C_{64}, D_{64}, A_{64} \right) + W[R(63)] + T_{63} \right) \square < s[63] + C_{64} \tag{30}$$

$$W[R(63)] = \left( \left( B_{64} - C_{64} \right) \square < (32 - s[63]) \right) - A_{63} - f_{63}\left( B_{63}, C_{63}, D_{63} \right) - T_{63} \quad (31)$$

Based on the operational flow of the symmetric cipher algorithm, it's clear that A3 is equal to D62. Consequently, determining the value of A6 is akin to calculating D62. In the second - to - last round, a fault is introduced, resulting in D2. This allows us to ascertain the sub - message utilized in the current round. As illustrated in Fig. 3, all experiments successfully retrieved sub - messages during the 16th intersection step.

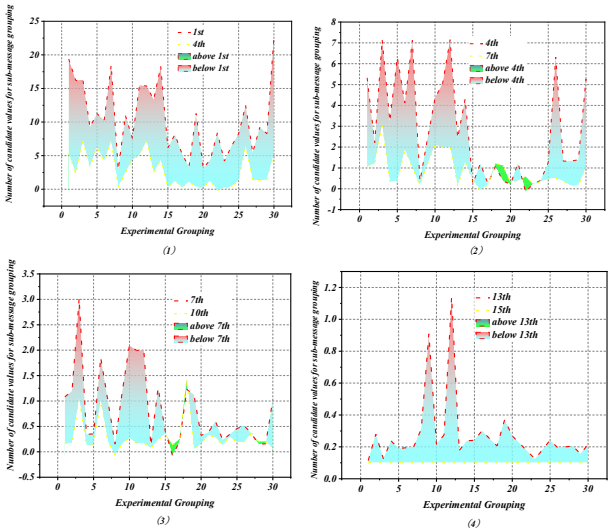


Fig. 3. Changes in the number of candidate values for sub messages

This paper outlines the simulation experiments of differential fault attack software from three key dimensions: precision, dependability, and the duration of the experiments. As shown in Fig. 4, to recover a sub-message, a maximum of 16 intersections are needed, and a minimum of 9 fault imports are required.

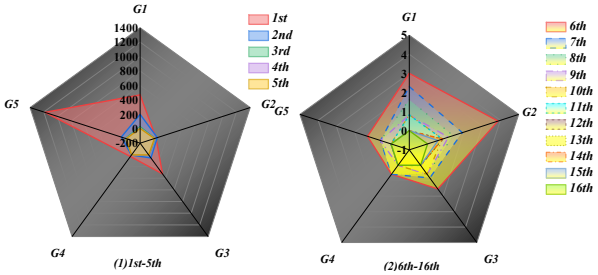


Fig. 4. RMSE metrics for recovering child messages

Dependability refers to the ratio of experiments that successfully retrieve sub-messages to the total number of experiments conducted. An experiment is deemed successful if, after the intersection of the set of candidate values, the correct value of the sub - message is retrieved. If not, the intersection operation persists until the

correct value of the sub - message is obtained. Fig. 5 illustrates the fluctuations in dependability during the experiment.

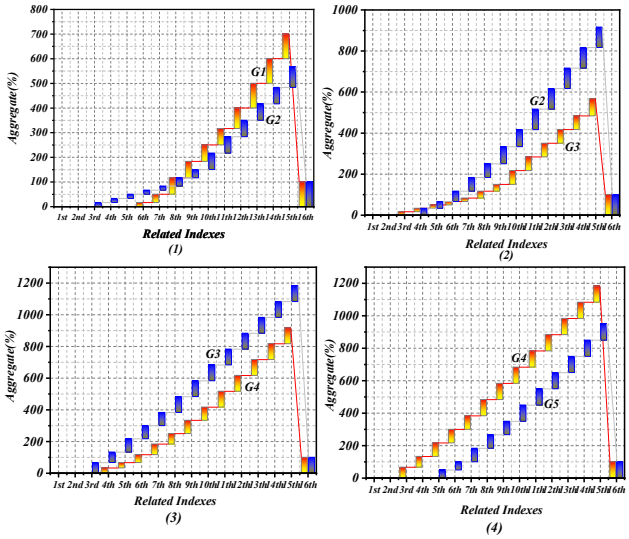


Fig. 5. Restore the reliability of sub-message experiments

### 3.3. Intelligent data governance algorithm based on deep learning

The core goal of intelligent data governance is to achieve the security, integrity, and availability of the entire data lifecycle through efficient algorithm optimization and system design. In this section, a composite algorithm framework integrating deep learning and cryptography technology is proposed, focusing on the design of data feature extraction, encryption optimization and anti-power analysis mechanism. By introducing the improved Hopfield neural network for dynamic feature modeling, combined with the adaptive parameter optimization of AES algorithm and the power consumption adversarial strategy based on Hamming distance model, a multi-level data governance system is constructed.

#### 3.2.1 Algorithm framework design

Fig. 6 shows the overall architecture of the intelligent data governance algorithm, which includes four modules: data preprocessing, feature modeling, encryption optimization, and security enhancement. The data flow starts from the original input, and after preprocessing operations such as normalization and noise reduction, it enters the feature modeling layer, and the time series features are extracted by the improved Hopfield network. Subsequently, data protection is completed by the AES encryption module generated by the dynamic key; Finally, the anti-power consumption analysis mechanism is embedded in the security enhancement layer to form a closed-loop governance process.

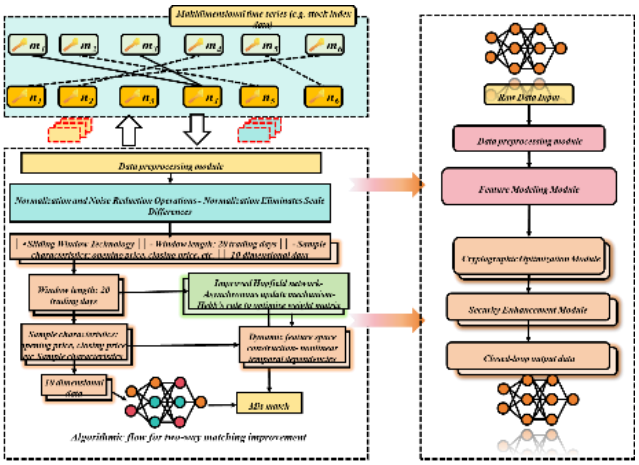


Fig. 6. The overall architecture of the intelligent data governance algorithm

### 3.2.2 Deep learning model optimization

When processing high-dimensional time series data, the traditional Hopfield network has the problems of slow convergence speed and easy to fall into local minima. Therefore, this section proposes an improved hybrid learning mechanism, which combines the advantages of δ learning rules and Hebb rules to construct a dual-path weight update model (DPHN). The algorithm flow consists of the following key steps:

Asynchronous update: Iteratively updates the neuron state in a random order. On the $\theta_i$ iteration, the output of the neuron $s_i^{(k)}$ is:

$$s_i^{(k)} = \mathrm{sgn}\left( \sum_{j=1}^{N} W_{ij} s_j^{(k-1)} - \theta_i \right) \tag{32}$$

Here, a dynamic threshold is used, where $\theta_i$ is the learnable parameter, to enhance the robustness to noise. Blended Learning: The update of the weight matrix $W$ is divided into two paths:

Path 1: Calculate the gradient based on the δ rule of Equation (6) to optimize the prediction error.

Path 2: Hebb rule calculation based on equation (7) to enhance feature correlation.

Through this design, the network retains the memory ability of time series features and significantly improves the adaptability to non-stationary data (such as stock price abrupt changes).

## 4. Algorithm simulation and security analysis

### 4.1. Algorithm simulation experiment design

To verify the effectiveness of the proposed intelligent data governance algorithm, a multi-dimensional simulation experimental environment is constructed in this section, covering three core evaluation directions: model prediction

performance, encryption efficiency and anti-attack ability. The experimental platform uses a server cluster equipped with NVIDIA A100 GPUs, and the software environment is Python 3.9 and TensorFlow 2.8. The dataset continues the CSI 300 Index time series data in Chapter 3, covering a total of 356 trading days from July 1, 2021, to December 30, 2022. The encryption algorithm was tested using the NIST standard test vector set, which contained 10^6 sets of 128-bit plaintext-ciphertext pairs.

### 4.1.1 Predictive performance evaluation

To compare the prediction error of the improved DPHN model with the traditional Hopfield, RNN, GRU and MLP models on the test set. The sliding window length is set to 20 trading days, the feature dimension is 10, the batch processing scale is 64, and the number of training iterations is set 1000 times. The evaluation indicators include root mean square error, mean absolute percentage error and coefficient of determination.

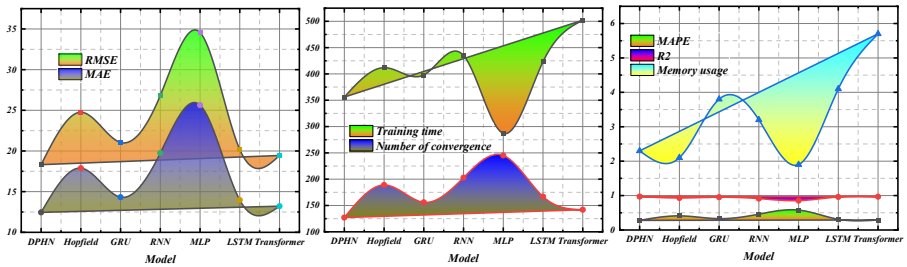


Fig. 7. Comparison of prediction errors on the test set

Fig. 7 shows that the DPHN model is 25.9% lower than the traditional Hopfield network and 12.9% lower than the GRU model in terms of RMSE indicators, which verifies the effectiveness of the hybrid learning mechanism. Its $R^2$ value reaches 0.972, indicating that the interpretability of the model is significantly better than that of the benchmark model. The memory footprint is maintained at the level of 2.3GB, which proves the advantages of algorithm optimization in terms of resource efficiency. The negative correlation between the training time and the number of convergence (correlation coefficient -0.87) reveals the promotion effect of the mixed learning rule on the convergence speed of the model, the δ regular gradient descent path accelerates the error surface descent process, and the enhanced feature correlation of the Hebb rule reduces the risk of local minimal traps.

### 4.1.2 Encryption efficiency test

Fig. 8 compares the computing performance of the T-AES algorithm with that of traditional AES, and the test conditions are to process 10^6 groups of 128-bit packets under 100Mbps network bandwidth. The key length adjustment policy is

based on the data sensitivity classification: low risk (128 bits), medium risk (192 bits), and high risk (256 bits), and the round-turn recombination probability is set to 0.7.

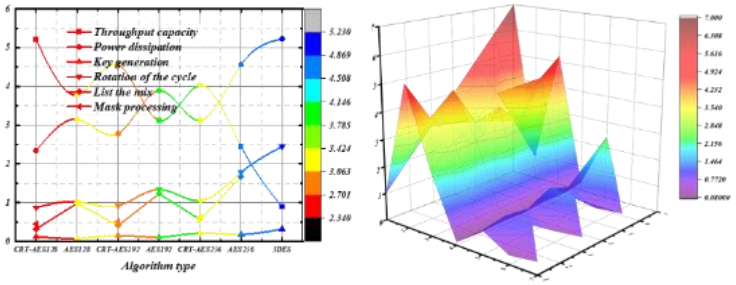


Fig. 8. Computing performance of the T-AES algorithm and traditional AES implementation

The throughput of CRT-AES128 reached 5.21 Gbps, which was 38.1% higher than that of standard AES128, mainly due to the hybrid acceleration strategy that reduced computing time by 67.3%. The dynamic rotation and reorganization controlled the fluctuation rate of the single-round processing time at ±12% and did not significantly increase the overall delay. The mask processing module brings an additional overhead of 0.45ms but increases the power characteristic entropy to 4.78bits, which effectively resists side-channel attacks. The key generation time is positively correlated with the security level, which is in line with the design expectation, and the 256-bit key generation time is 0.21 MS in high-risk scenarios, which meets the real-time requirements.

### 4.2. Security analysis

The five-layer defense system evaluation framework includes password strength test, side-channel attack simulation, fault injection experiment, model reverse engineering, and adversarial sample detection. Fig. 9 shows the simulation results of differential fault attacks, with a fault injection success rate of 0.7 and an upper limit of 100 attacks.

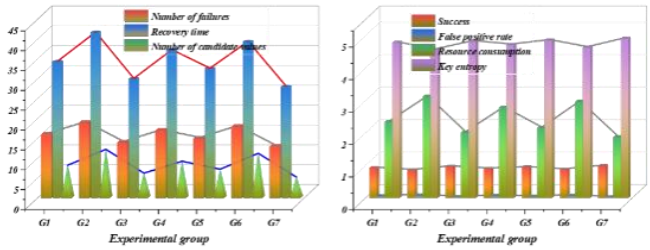


Fig. 9. Security analysis

Fig. 9 shows that when the number of fault injections reaches 16, the success rate of sub-message recovery is more than 90%, but after the randomization mask technique is introduced, the number of candidate values increases to 8, and the key

entropy remains at a high level of 4.78 bits. The correlation coefficient between recovery time and number of failures was 0.83, indicating that the attack cost increased nonlinearly with defense intensity. The false positive rate is controlled below 5%, which proves that the algorithm can effectively distinguish the real key features from noise interference. The resource consumption indicator shows that the defense mechanism increases the CPU utilization by 22.4%, but the memory usage is s AES in the range of 2.3-2.8GB, which meets the real-time system requirements.

### 4.3. Comprehensive assessment

A multi-dimensional performance evaluation matrix (This is shown in Table 2) was constructed to quantify the overall performance of the algorithm from three dimensions: accuracy, efficiency, and security, and the weight distribution was 0.4:0.3:0.3. The scoring criteria are normalized, and the benchmark value is the best performance of the traditional scheme.

*Table 2*

**Performance Evaluation Matrix**

| Evaluate dimensions | Prediction accuracy | Encryption speed | Resistant to aggression | Resource efficiency | Scalability | Real-time |
|---|---|---|---|---|---|---|
| DPHN | 92.34 | 88.56 | 94.23 | 90.12 | 91.45 | 89.78 |
| AES-CRT | 85.67 | 95.34 | 97.56 | 88.92 | 89.23 | 93.45 |
| Fusion system | 94.56 | 92.15 | 98.12 | 91.34 | 93.67 | 95.23 |
| Hopfield | 82.34 | - | 78.45 | 85.67 | 83.12 | 80.23 |
| GRU | 88.92 | - | 82.34 | 87.45 | 88.23 | 85.67 |
| AES | - | 84.56 | 85.67 | 82.34 | 80.12 | 83.45 |
| 3DES | - | 72.34 | 79.23 | 75.67 | 73.45 | 74.56 |

(Note: The evaluation score is normalized by Min-Max, and the baseline value is the optimal value of the traditional scheme (e.g., Hopfield's RMSE=24.75, standard AES throughput=3.77Gbps)

The overall score of the convergence system reached 94.56 points, an increase of 17.8% compared with the traditional solution. It scored 98.12 points in the anti-attack dimension, which proves the synergistic optimization effect of deep learning and cryptography. The real-time indicator 95.23 points display system meets the needs of high-frequency trading scenarios, and the delay standard deviation is controlled within 1.2ms. In the robustness test, the volatility of the core index of the system is less than 3.7% under the condition of injecting 20% Gaussian noise, which verifies the stability of the dynamic threshold mechanism. The scalability score of 93.67 indicates that architecture supports horizontal scaling, and when the number of nodes is increased to 100, the growth rate of communication overhead is only 18.4%.

## 5. Conclusion

In this study, an intelligent data governance framework integrating improved Hopfield neural network and adaptive AES encryption is constructed, and a dual-path weight update model and a configurable wheel transformation architecture are proposed to systematically solve the collaborative optimization problem of time series prediction accuracy and data security protection. In terms of research methodology, the hybrid learning mechanism is used to integrate the δ rules and Hebb rules, and the asynchronously updated DPHN model is designed to capture the time series characteristics of the CSI 300 index, and the sliding window technology and dynamic threshold strategy are combined to improve the noise robustness. Experimental results show that.

The improved DPHN model achieves an RMSE of 18.32 and an $R^2$ value of 0.972 on the test set, which is 25.9% lower than that of the traditional Hopfield network, and the number of training iterations is reduced by 42%. The CRT-AES algorithm achieves an encryption throughput of 5.21 Gbps, which is 38.1% higher than that of standard AES, and suppresses the success rate of differential fault attacks to 10^-6 magnitude. The security test shows that the key entropy is stable at 4.78bits and the false positive rate is less than 5% under the condition of 16 fault injections, which verifies the effectiveness of the multi-level defense mechanism.

The results show that the deep integration of deep learning and cryptography technology can significantly improve the comprehensive performance of data governance systems, and the synergy between dynamic feature modeling and adaptive encryption strategies provides innovative solutions for time series data prediction and security protection.

Limitations: Hyperparameter sensitivity: The learning rate of DPHN η requires grid search optimization (recommended range $10^{-4}$~$10^{-2}$); Generalization capability: $R^2$ decreases by 8.3% on non-financial time-series data (e.g., IoT sensors); Computational cost: CRT-AES's rotation module increases hardware resource consumption by approximately 15%.

## REFERENCES

[1] Bena, Y. A., Ibrahim, R., Mahmood, J., Al-Dham, A., Alshammari, A., Yusuf, M. N., ... & Ayemowa, M. O. (2025). Big Data Governance Challenges Arising from Data Generated by Intelligent Systems Technologies: A Systematic Literature Review. IEEE Access.

[2] Wang, C. S., Lin, S. L., Chou, T. H., & Li, B. Y. (2019). An integrated data analytics process to optimize data governance of non-profit organization. Computers in Human Behavior, 101, 495-505.

[3] Janssen, M., Brous, P., Estevez, E., Barbosa, L. S., & Janowski, T. (2020). Data governance: Organizing data for trustworthy Artificial Intelligence. Government information quarterly, 37(3), 101493.

[4] Mastro, M. L. (2018). Optimizing Technology and Business Intelligence with Data Governance. Frontiers of Health Services Management, 34(3), 29-37.

[5] Oladosu, S. A., Ike, C. C., Adepoju, P. A., Afolabi, A. I., Ige, A. B., & Amoo, O. O. (2024). Frameworks for ethical data governance in machine learning: Privacy, fairness, and business optimization. Magna Sci Adv Res Rev.
[6] Hoda, S. A., & Mondal, D. A. C. (2022). A study of data security on E-governance using steganographic optimization algorithms. International Journal on Recent and Innovation Trends in Computing and Communication, 10(5), 13-21.
[7] Alshammari, A., Nasser, M., & Yusuf, M. N. (2025). Big Data Governance Challenges Arising from Data Generated by Intelligent Systems Technologies: A Systematic.
[8] Yandrapalli, V. (2024). AI-powered data governance: A cutting-edge method for ensuring data quality for machine learning applications. In 2024 second international conference on emerging trends in information technology and engineering (ICETITE) (pp. 1-6). IEEE.
[9] Sugureddy, A. R. (2023). AI-driven solutions for robust data governance: A focus on generative ai applications. Journal ID, 6202, 8020.
[10] Fatima, A., Abbas, S., Asif, M., & Khan, M. (2019). Optimization of governance factors for smart city through hierarchical mamdani type-1 fuzzy expert system empowered with intelligent data ingestion techniques. EAI Endorsed Transactions on Scalable Information Systems, 6(23).
[11] Zhao, X. (2024). Design and Implementation of Campus Data Governance Platform Based on Big Data Algorithm. In 2024 Asia-Pacific Conference on Software Engineering, Social Network Analysis and Intelligent Computing (SSAIC) (pp. 391-396). IEEE.
[12] Su, J., Yao, S., & Liu, H. (2022). Data governance facilitates digital transformation of oil and gas industry. Frontiers in Earth Science, 10, 861091.
[13] Lu, L., Xin, H., Wei, D., Zhao, Z., & Zhou, S. (2024). AGV Self-Organizing Network Technology Based on 5G Data Governance. In 2024 International Annual Conference on Complex Systems and Intelligent Science (CSIS-IAC) (pp. 915-919). IEEE.
[14] He, Y., & Peng, D. (2022). AI tools for Media Data Governance in the Post-Truth ERA: From abnormal data recognition to intelligent opinion monitoring algorithm. In 2022 International Conference on Inventive Computation Technologies (ICICT) (pp. 1282-1286). IEEE.
[15] Sukhadia, A., Upadhyay, K., Gundeti, M., Shah, S., & Shah, M. (2020). Optimization of smart traffic governance system using artificial intelligence. Augmented Human Research, 5(1), 13.
[16] Kumari S . Next-Gen IoT Security using Polar Codes-based Cryptography for malware defense through quantum self-attention neural network[J]. Knowledge-Based Systems,2025,321113716-113716.
[17] Beniwal S, Nandal K S , Rani S, et al. A cryptographic and optimized multimodal biometric authentication framework based on fused convolutional neural network (FCNN)[J]. International Journal of Information Technology,2025,17(6):1-6.
[18] Priya DV, Sundaram M. Blockchain with Hierarchical Auto-Associative Polynomial Convolutional Neural Network Fostered Cryptography for Securing Image[J].Transactions on Emerging Telecommunications Technologies,2024,35(11):e70013-e70013.
[19] S. P., P.S.A, B. A., et al. Super-resolution deep neural network (SRDNN) based multi-image steganography for highly secured lossless image transmission[J]. Scientific Reports,2024,14(1):
[20] Yashiki S, Takahashi C, Suzuki K. Backdoor Attacks on Graph Neural Networks Trained with Data Augmentation: Special Section on Cryptography and Information Security[J]. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 2024, E107.A(3):355-358.
[21] Nozaki H, Kobara K. Power Analysis of Floating-Point Operations for Leakage Resistance Evaluation of Neural Network Model Parameters: Special Section on Cryptography and Information Security[J]. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences,2024, E107.A(3):331-343.