World Journal of Information Technology
ISSN: 2959-9911
DOI: https://doi.org/10.61784/wjit3036

DEEP LEARNING-ENHANCED DYNAMIC FRAME SLOTTED ALOHA OPTIMIZATION ALGORITHM

HongLing Zhang
School of Information Engineering, Zhongyuan Institute Of Science And Technology, Xuchang 461113, Henan, China.
Corresponding Email:19837698961@163.com

Abstract: In recent years, Radio Frequency Identification (RFID) technology has seen expanding applications across both industrial production and daily life, driving growing demand for efficient tag reading systems.When faced with a large numeral of tags, the Dynamic Framed Slotted ALOHA (DFSA) algorithm keeps the throughput at a high position for most of the time. The chief principle is to dynamically adjust the frame length based on the response consequence of the triumphant transmission of the earlier frame, so that during the data container transmission process, the length of each frame is close to the number of data packets that need to be transmitted within the range that the information transmission system can receive. Based on the analysis of traditional algorithms, this paper proposes a new tag number estimation algorithm. This algorithm based on Transformer, residual connections, and Multi-Layer Perceptron (MLP), combined with algorithms such as tag grouping techniques. Compared with traditional algorithms, the algorithm proposed in this paper addresses the shortcomings of the traditional ALOHA protocol in dynamic frame slot adjustment, collision avoidance, and throughput optimization by introducing deep learning. It significantly improves the effectiveness and reliability of RFID systems and is able of maintaining a high data storage rate even in scenarios with large amounts of data.
Keywords: RFID; DFSA; MLP-Transformer

1 INTRODUCTION

RFID is a non-contact automatic discovery technology that uses radio recurrence signals to automatically recognize targets and retrieve associated data. RFID systems are widely applied in logistics, manufacturing, healthcare, and other fields, their performance is often hindered by tag collision problems, which significantly reduce identification efficiency[1-8]. ALOHA-based algorithms are widely adopted due to their simplicity, but they suffer from low throughput and high collision rates under heavy tag loads. In contrast, binary tree algorithms offer better collision resolution but are computationally complex, making them impractical for real-time systems.To solve these limitations, this paper proposes an improved ALOHA algorithm that dynamically adjusts frame time slots using deep learning. Our method enhances identification efficiency by predicting optimal slot allocation, reducing collisions, and maintaining low computational overhead. This innovation bridges the gap between performance and practicality in large-scale RFID deployments. Among these, research on the performance of RFID anti-collision algorithms [9] is particularly remarkable. Currently, widely used anti-collision identification algorithms can be divided into inflexible allowance types and dynamic allocation types. Fixed allocation types include Time Division Multiple Access (TDMA), Frequency Division Multiple Access (FDMA), and Code Division Multiple Access (CDMA). Dynamic allocation types can be further categorized into contention-based access, reservation-based access, and polling-based access.

In RFID systems, the primary approaches to address tag collision issues fall into two distinct categories: predetermined channel assignment methods and adaptive channel contention strategies. The former involves preallocating dedicated communication resources to individual tags, eliminating the unpredictability associated with contention-based mechanisms. Nevertheless, this approach demonstrates inefficiency in resource utilization during periods of low activity and exhibits limited flexibility when confronted with unexpected service demands. On the other hand, adaptive contention strategies enable tags to opportunistically access available channels according to established contention parameters, offering enhanced responsiveness to fluctuating tag populations. Within this category, protocols derived from the ALOHA framework have gained widespread adoption owing to their straightforward implementation and seamless integration with existing system architectures. Regarding tag population estimation, conventional approaches encompass the Q-value assessment technique, the Schouten analytical method, and the Vogt iterative algorithm. The Q-value approach determines tag quantities by monitoring response attempts, providing computational simplicity at the expense of precision. Schoute's methodology, integrated with the Dynamic Frame Slotted ALOHA framework, approximates tag populations by analyzing collision slot proportions. The Vogt algorithm, while achieving optimal theoretical accuracy through iterative comparison between empirical observations (including empty slots, successful transmissions, and collision events) and their mathematical expectations, suffers from substantial computational overhead. This characteristic notably diminishes operational effectiveness, especially in scenarios involving large-scale tag deployments.

Traditionally, Deep learning techniques have been widely used in such problems. Reference [10] is based on the energetic mounting slotted algorithm and BP neural network. By processing the dataset of tag quantity, it establishes the mapping relationship between the reader and the tag quantity. This algorithm improves the system efficiency without sacrificing its accuracy[10]. Reference [11] is based on the dynamic frame slotted ALOHA algorithm (D-G-MFSA)

with tag grouping and long short-term memory (LSTM). It regards the tag quantity as a time series, uses LSTM for real-time prediction, dynamically adjusts the frame length, and groups tags when the tag quantity is large. This algorithm promotes the stability of the system when the tag quantity is large, and has the advantages of simple principle and high reading efficiency[11]. Reference [12] combines the dynamic frame slotted ALOHA (DFSA) algorithm with Transformer and long short-term memory (LSTM) neural networks. This algorithm ensures more accurate tag quantity prediction, reduces the time consumption of the reading system, and improves the system throughput[12].

This paper trains a neural network based on residual connections, a multi-layer perceptron (MLP), and a Transformer encoder, combined with the tag grouping algorithm, to predict the number of tags.The main contribution of this paper is as follows:

(1) Residual connections accelerate training convergence, and the parallel computing advantages of Transformer improve computational efficiency.

(2) By incorporating tag grouping, the algorithm achieves scalability for high-density tag environments.

(3) Experimental results indicate that, compared to the traditional BP network algorithm, the proposed manner accomplish higher throughput and consumes fewer total time slots.

(4) MLP is combined with residual connections, it maintains stable gradient norms during backpropagation, resulting in a 3x faster training convergence speed compared to traditional BP networks.

(5) The DFSA algorithm's dynamic frame length adjustment mechanism estimates the number of tags in real-time and automatically optimizes the frame size, improving the theoretical system throughput. Additionally, its time slot grouping technique further reduces collision probability.

2 DFSA ALGORITHM

2.1 Algorithm Principle

The pure ALOHA protocol serves as a fundamental multiple access control mechanism in wireless communication systems.The ALOHA protocol is characterized by the fact that any station can transmit immediately after a slot is generated and determines whether the transmission is successful by detecting signal feedback on the channel. When two or more stations transmit data simultaneously, a collision occurs, resulting in a high collision rate and low efficiency. SALOHA improves the throughput of the ALOHA system by dividing time into fixed slots, where tags can only transmit data at the beginning of a slot, thereby reducing collisions. However, the frame length is fixed, leading to limited resource utilization. The FSA protocol further enhances the time division dimension. The DFSA protocol dynamically adjusts the frame length based on the successful transmission responses from the previous frame, ensuring that the length of each frame is close to the optimal value during data packet transmission. This significantly improves throughput and resource utilization.

2.2 Mathematical Analysis

The fundamental principle of dynamically modifying the frame length in DFSA (Dynamic Frame Slotted ALOHA) lies in establishing the correlation between the current number of unidentified tags and the most efficient frame size. In this study, we define the frame length as **N**, while the total number of tags within the reader's detection range is denoted as **M**. Each tag has an equal and independent probability of selecting any given time slot within the frame. This assumption ensures that tag transmissions are uniformly distributed across the available slots, facilitating effective collision management and frame length optimization. Let its probability be $\mathrm{P}=\frac{1}{\mathrm{L}}$, Thus the quantity of arriving tags X in each time slot conforms to the binomial distribution $\mathrm{X}\sim\mathrm{B}(\mathrm{N},\frac{1}{\mathrm{L}})$, Then the probability of having a tag in a time slot is:

$$P_x=\binom{k}{x}\left(\frac{1}{M}\right)^x\left(1-\frac{1}{M}\right)^{k-x} \tag{1}$$

Where there are no tags in the time gap, the corresponding probability is as follows ($\mathrm{P_{idle}}$), When there is a single tag in the time gap, the corresponding probability is as follows ($\mathrm{P_{succ}}$). When there are multiple tags in the time gap, the corresponding probability is as follows ($\mathrm{P_{coll}}$). These probabilities can be expressed as:

$$P_{\text{idle}}=\left(1-\frac{1}{M}\right)^k \tag{2}$$

$$P_{\text{succ}}=k\times\frac{1}{M}\times\left(1-\frac{1}{M}\right)^{k-1} \tag{3}$$

$$P_{\text{coll}}=1-P_{\text{idle}}-P_{\text{succ}} \tag{4}$$

The expected values for the number of idle time slots, successful time slots, and collision time slots within a frame can be derived based on the given parameters. Let the frame length be N and the total number of tags be M. Assuming each tag independently and uniformly selects a time slot, the probabilities and expected values can be calculated as follows:

$$y_{\text{idle}}=k\times P_{\text{idle}} \tag{5}$$

$$y_{\text{succ}}=k\times P_{\text{succ}} \tag{6}$$

$$y_{\text{coll}}=k\times P_{\text{coll}} \tag{7}$$

Clearly, the estimated number of remaining unrecognized tags $k_{estimate}$ is:

$$k_{\text{estimate}} = k - y_{\text{succ}} \tag{8}$$

According to the formula, when the number of tags k is 1-4, a frame length M=4 is sufficient to cover the number of tags and reduce collisions. When the number of tags k>4, using M=4 would lead to increased collisions, thus requiring a larger frame length. When the number of tags k is between 5 and 10, a frame length M=8 can effectively reduce collisions. That is, as the number of tags k increases, the frame length gradually increases. The theoretical analysis and experimental validation results based on the dynamic frame slotted algorithm are shown in Table 1.

Table 1 Relationship between frame size and label number of dynamic frame time slot algorithm

Frame length	4	8	16	32	64	128	256
Number of tags	1~4	5~10	11~22	23~44	45~88	89~177	Above 177

2.3 Tag grouping Algorithm

The Tag Grouping Algorithm is a technique used in Radio Frequency Identification (RFID) systems, aiming to optimize the label discovery process by grouping labels, thereby reducing collisions and improving system effectiveness. Formula (9) is used to calculate the number of tags after grouping, where a and b are the number of tag groups, and N is the total number of tags. This formula aims to determine the minimum and maximum number of tags in each group under different grouping numbers. For different grouping numbers, the minimum number of tags is obtained by dividing the total number of tags by the number of groups and rounding down, while the maximum number of tags is obtained by dividing the total number of tags by the number of groups and rounding up.

$$\frac{\frac{N}{a}}{256} \times \left(1 - \frac{1}{256}\right)^{\frac{N}{a}-l} = \frac{\frac{N}{b}}{256} \times \left(1 - \frac{1}{256}\right)^{\frac{N}{b}-l} \tag{9}$$

When a = 1 and b = 2, substituting into formula (9) yields a critical value N of 355.With 2 groups, the tag capacity increases to 356-709. By changing the values of parameters a and b in formula (9), different critical values for tag grouping are obtained,this demonstrates the scalable nature of the grouping algorithm, where doubling the group count approximately doubles the maximum tag capacity, maintaining consistent range intervals of ~355 tags per grouping threshold. The results are shown in Table 2.

Table 2 The Correlation between the Count of Groups and the Quantity of Labels

Number of groups	1	2	4	8	...
Minimum number of tags	1	356	710	1418	...
Maximum number of tags	355	709	1417	2834	...

3 METHOD

The Transformer model is a deep learning architecture. The Transformer model is a deep learning model based on the attention mechanism, abandoning the use of CNN and RNN in previous deep learning tasks. This model exclusively utilizes the attention mechanism to model global relationships between inputs and outputs. Unlike traditional recurrent neural networks (RNNs) that process data sequentially, it enables parallel processing of sequence data while effectively capturing long-distance dependencies through its attention-based architecture.

The calculation for dot-product attention is as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{10}$$

Where Q is the query matrix, K is the key matrix, V is the value matrix, and dk is the dimension of the keys.

The multi-head attention mechanism divides the input into multiple parallel attention heads.Each head computes scaled dot-product attention independently, then the outputs are concatenated and linearly transformed. The computation for each head is given by:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{11}$$

$$\text{head}_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{12}$$

This allows the model to focus on different subspaces and capture more features and information.

This paper also uses residual connection. The core idea is: instead of learning an underlying mapping (H(x)) directly, the network learns the residual (F(x) = H(x) - x). This makes it easier to optimize because fitting a residual to zero (for identity mapping) is simpler than learning a new function from scratch, especially for deeper layers.

The input x undergoes dimensionality expansion through a completely linked layer. The MLP module processes the features via two fully connected layers (with ReLU activation), and the output is added to implement a residual connection. This feature-enhanced data is then fed into the core Transformer encoder (a 2-layer stacked structure) where each layer consists of a multi-head self-attention mechanism for capturing dependencies between time slots and a position-wise feedforward network. Throughout the Transformer processing, the tensor dimension remains (batch_size,

sequence_length, hidden_size). Finally, the encoder output is flattened and projected through a fully connected output layer to obtain a prediction result with a dimension of 1.The transformer processes the self-attention weight matrix to explicitly capture temporal slot correlation patterns, while the multi-head mechanism learns multiple dependency relationships in parallel, improving the accuracy of label quantity prediction.The flowchart is shown as Figure 1.

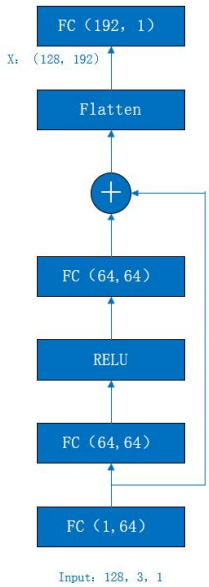

Figure 1 Neural Network Structure

The specific steps of the proposed algorithm are as follows:
1. Data Calculation and Preparation
Using Equations (5) to (8), compute the number of idle slots y_{idle}, successful slots y_{succ}, collision slots y_{coll}, and the estimated number of remaining tags $k_{estimate}$ for different tag quantities k (assuming $k \in [0,\ldots,5M]$) and frame lengths M. These results are compiled into the datasets $T = \{(k, M, y_{idle}, y_{succ}, y_{coll}) | k_{estimate}\}$.
2. Network Training
The datasets T is fed into the network for training. Once the network metrics converge, the corresponding network model G for each frame length M is obtained.
3.Real-Time Prediction and Adjustment
During the reader's current reading cycle, the actual detected number of tags (k), idle slots y_{idle}, successful slots y_{succ},and collision slots y_{coll} are input into the corresponding network model G to predict the remaining number of tags $k_{estimate}$. The reader then dynamically adjusts the frame length for the next reading cycle based on the rules in Table 1.

4 TRANSFORMER-OPTIMIZED DFSA ALGORITHM

4.1 Model Training

During the model training phase, we adopted the following configurations: the parameter update step size (learning rate) was set to 0.01, with the maximum number of training epochs limited to 150. To enhance the model's nonlinear representation capability, we employed the ReLU (Rectified Linear Unit) activation function in the network. Other key hyperparameter settings included: 150 iterations, Lr regularization coefficient of 0.001, batch size of 128, the Adam optimizer algorithm, and Mean Squared Error (MSE) as the loss function for optimization objectives.
Finally, as shown in Figure 2, the training results indicate that:
Initial Phase (Epoch 0–25): Both the train loss (blue) and test loss (orange) start high but drop sharply. This rapid decrease indicates that the model quickly learns essential patterns in the early training stages.Mid - Training Phase (Epoch 25 – 150): The losses continue to decline but with increased fluctuations. These variations arise from the stochastic nature of batch - based training—each batch's unique data introduces minor instability. Despite this, both curves trend downward, showing consistent learning.Final Phase (Epoch 150–200): By epoch 200, both losses stabilize near \(10^{-3}\). The close alignment of train and test loss throughout training suggests no significant overfitting. If overfitting occurred, the test loss would diverge upward while the train loss continued decreasing. Here, their proximity indicates the model generalizes well to unseen data. Overall, the curve demonstrates effective training: rapid initial learning, steady improvement, and good generalization, as evidenced by the parallel trends of train and test loss.

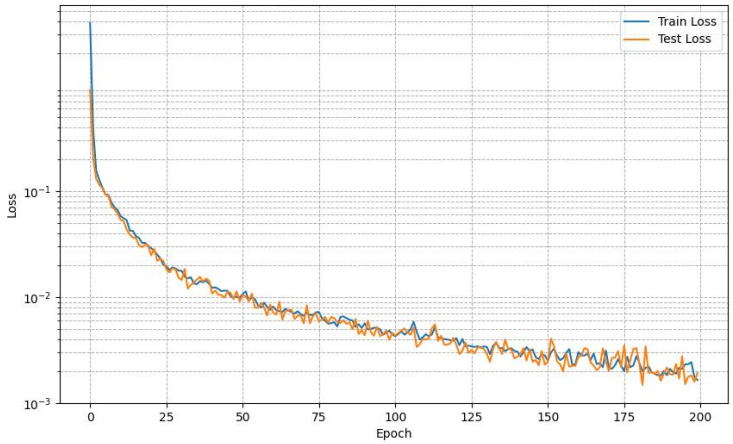

Figure 2 Training Loss with Epoch

4.2 Algorithm Performance Analysis

The primary metrics for evaluating tag reading efficiency consist of two key dimensions.The definition of system throughput is the ratio of the number of successfully identified tags to the total number of slots consumed in a single read cycle.In Radio Frequency Identification (RFID) systems, throughput typically refers to the number of tags successfully read per unit of time.，reflecting the efficiency and capability of the system in processing data. As shown in Figure 3, as the number of tags increases, the system throughput initially rises and then declines. The throughput reaches its peak when the number of tags is around 200. After this peak, as the number of tags continues to increase, the system throughput shows a downward trend. Following the peak throughput, the decline in throughput is significantly faster in the BP algorithm compared to the TLE algorithm as the number of tags increases. When the number of tags reaches around 1200, the throughput in the TLE algorithm begins to increase again, until the number of tags reaches around 1400, after which the throughput starts to decline once more.

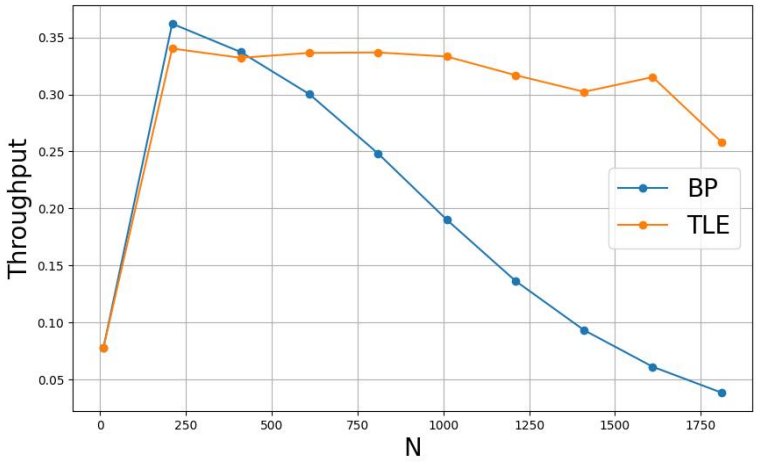

Figure 3 Comparison of the Throughput

As shown in Figure 4, the number of slots consumed by the algorithm proposed in this paper is lower than that of the traditional BP neural network algorithm. The total number of consumed slots refers to the total number of slots used for tag identification in a single read cycle of a Radio Frequency Identification (RFID) system. It is one of the key metrics for measuring system resource consumption, directly impacting the system's throughput and efficiency.As shown in Figure 4, when the number of tags increases, the total number of consumed slots for both algorithms also increases. When the number of tags is less than 600, the total number of consumed slots for both algorithms is approximately the same. However, when the number of tags exceeds 600, the total number of consumed slots for the BP algorithm increases significantly faster than that for the TLE algorithm.

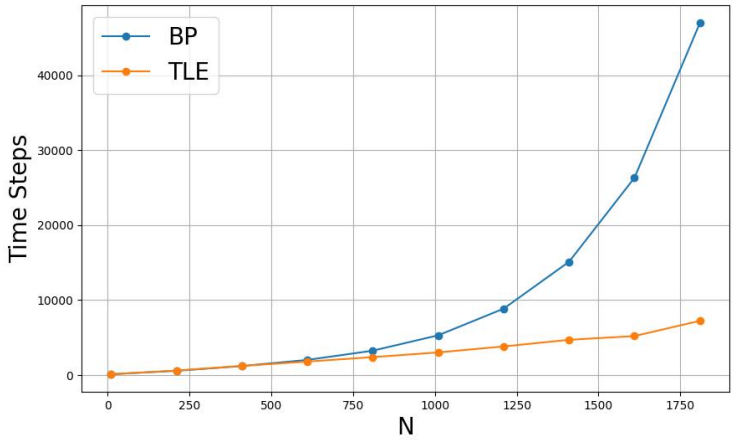

Figure 4 Comparison of the Number of Consumed Slots

5 CONCLUSIONS

This study proposes an RFID tag identification optimization algorithm based on a Transformer architecture. By introducing a hybrid "Transformer + Residual MLP" encoding structure, the multi-head self-attention mechanism explicitly models the nonlinear relationships between tag time slots, while residual connections enhance training convergence speed. Compared with conventional algorithms, the proposed deep learning-based approach addresses key limitations of traditional ALOHA protocols in dynamic frame slot adjustment, collision avoidance, and throughput optimization, significantly improving the efficiency and reliability of RFID systems. The algorithm maintains high data storage rates even in large-scale data scenarios, providing an effective solution for performance enhancement in high-demand RFID applications.

With the advancement of IoT technology, RFID performance demands are growing. The algorithms in this paper offer robust technical support for logistics management and other fields. By combining dynamic frame length optimization with deep learning, the TLE algorithm overcomes traditional RFID efficiency bottlenecks and demonstrates cross-scenario adaptability. Future research will focus on model lightweighting, transfer learning, and multi-protocol fusion to enhance real-time responsiveness and generalization in edge computing and high-collision scenarios, driving the evolution of next-generation IoT ecosystems.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] Maksimenko A, Dobrykh D, Yusupov I, et al. Miniaturization limits of ceramic UHF RFID tags. Scientific Reports, 2025, 15(1): 10984.
[2] Masekwana F, Jokonya O. Factors affecting the adoption of RFID in the food supply chain: A Systematic Literature Review. Frontiers in Sustainable Food Systems, 2025, 8: 1497585.
[3] Claucherty E, Cummins D, Aliakbarian B. RFID Unpacked: A Case Study in Employing RFID Tags from Item to Pallet Level. Electronics, 2025, 14(2): 278.
[4] Wu Y, Lin J, Chen H, et al. A transformer-based double-order RFID indoor positioning system. Expert Systems with Applications, 2025: 126530.
[5] Maimouni M, Abou El Majd B, Bouya M. Optimising RFID network planning problem using an improved automated approach inspired by artificial neural networks. Information Sciences, 2025: 121927.
[6] Lasantha L, Ray B, Karmakar N. Trade-Off Analysis for Array Configurations of Chipless RFID Sensor Tag Designs. Sensors, 2025, 25(6): 1653.
[7] Wang Z, Mao S. Generative AI-Empowered RFID Sensing for 3D Human Pose Augmentation and Completion. IEEE Open Journal of the Communications Society, 2025.
[8] Lazaro A, Cujilema M R, Villarino R, et al. A novel approach for wine anti-counterfeiting using laser-induced graphene chipless RFID tags on cork. Scientific Reports, 2025, 15(1): 12750.
[9] Liu H, Meng Z, Li C, et al. Modeling for Phase Decoupling to Detect the Orientation and Position of Moving Objects with Simple RFID Array. IEEE Transactions on Instrumentation and Measurement, 2025.
[10] Ming Dongyue, Wang Shangpeng, Lei Ming, et al. Improved Dynamic Framed Slotted ALOHA Algorithm Combined with BP Neural Network . Mini-Micro Systems, 2021, 42(09): 1920-1923.

[11] Li Z, Li Z. An RFID anti-collision algorithm integrated with LSTM. 2024 IEEE 4th International Conference on Power, Electronics and Computer Applications (ICPECA), IEEE, 2024: 985-989.
[12] Zhou Q. ALOHA Improvement Algorithm for Dynamic Frame Time Slots with Deep Learning. 2024 IEEE 4th International Conference on Data Science and Computer Application (ICDSCA), IEEE, 2024: 671-676.