Proceedings of the 2023 International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/32/20230186

Exploring the application and performance of extended hamming code in IoT devices

Liuxu Shen

Computer Science & Technology, Nanjing University of Information Science & Technology, Nanjing, 210044, China

202083420072@nuist.edu.cn

Abstract. This study primarily focuses on the implementation of extended Hamming code within Internet of Things (IoT) devices and examines its impact on device performance, particularly in relation to communication protocols. The research begins by introducing and explaining the essential principles surrounding the extended Hamming code and its system. This introduction is followed by a detailed analysis of its practical application in IoT device communication and the subsequent influence on performance. Additionally, the study explores the potential role of extended Hamming code in strengthening the security measures of IoT devices. Experimental findings indicate that incorporating extended Hamming code can effectively enhance the communication efficiency of IoT devices, ensuring accurate data transmission. It also improves the overall operational efficiency of the devices and fortifies their security framework. Yet, despite these promising outcomes, the real-world application of extended Hamming code presents significant challenges. These hurdles highlight the need for continued research and exploration to maximize the potential of the extended Hamming code in the IoT domain. The study concludes with an optimistic outlook, encouraging ongoing investigation and innovation to further optimize the benefits of this code and drive advancements in IoT technology.

Keyword: internet of things, extended hamming code, communication protocol, device performance, device security.

1. Introduction

The evolution of Internet of Things (IoT) technology is dramatically reshaping our daily lives. IoT devices, as essential elements of the IoT landscape, communicate using a variety of protocols to enable smart interaction between devices [1]. However, interruptions in the communication process can significantly compromise the efficiency and accuracy of data in IoT devices. The traditional Hamming code, a well-known tool for error detection and correction, is widely employed in various communication systems. Still, due to its limitations in certain demanding application scenarios, a more advanced solution—namely, the extended Hamming code—is required to meet stricter performance demands [2]. This paper delves into the implementation of the extended Hamming code within IoT devices, focusing on its impact on device communication protocols. The extended Hamming code's role in enhancing the communication efficiency of IoT devices, guaranteeing precise data transmission, and improving overall device performance is scrutinized. Additionally, the potential of the extended Hamming code to bolster the security measures of IoT devices is explored.

DOI: 10.54254/2755-2721/32/20230186

2. Related theories

2.1. Extended hamming code definition

Hamming code, a highly regarded method of error detection and correction, uses three check bits to encode four data bits, exemplified in the (7,4) Hamming code [3]. These check bits are strategically positioned at powers of two, and are generated using XOR operations, ensuring full positional coverage for error detection and correction [4].

Nevertheless, traditional Hamming code can stumble when confronted with multi-bit errors, like two-bit errors. This is because the error checking code can overlap with the one produced during a single-bit error, making it difficult to discern whether it's a single or double-bit error [5]. To combat this, an enhanced Hamming code has been proposed. This version appends an overall check bit, referred to as 'Pa', to the conventional Hamming code, and introduces an overall check code, named 'Ga'. These are both derived from specific XOR operations [6]. The enhanced Hamming code has the capacity to both detect and correct single-bit errors, as well as detect double-bit errors [7]. By analyzing the values of 'Pa' and 'Ga', it can distinguish between error types, thus enabling appropriate corrections. Therefore, this enhanced Hamming code holds significant potential for application within the communication protocols of IoT devices. As shown in Figure 1.

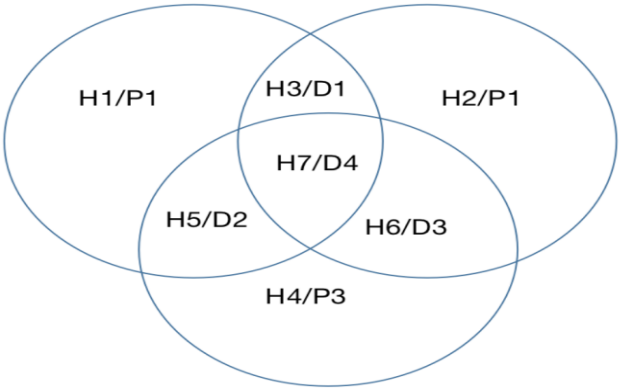

Figure 1. Schematic diagram of the intersection of each position of the hamming code (photo/picture credit: original).

$$G_1 = P_1 \oplus D_1 \oplus D_2 \oplus D_4 \quad (1)$$

$$G_2 = P_2 \oplus D_1 \oplus D_3 \oplus D_4 \quad (2)$$

$$G_3 = P_3 \oplus D_2 \oplus D_3 \oplus D_4 \quad (3)$$

2.2. Expand hamming code code system

In the context of deep space communication, the utilization of the extended Hamming code system is principally showcased through its robust capabilities of error detection and correction. Deep space communication is characterized by extreme distances, substantial signal degradation, and significant environmental noise, leading to a transmission error rate significantly higher than in typical communication scenarios [8]. This necessitates the adoption of a robust coding system to ensure communication reliability, a role fulfilled by the extended Hamming code.

At the transmission end of deep space communication, original data is initially encoded into the extended Hamming code. Each data bit generates corresponding check bits, and an additional parity check bit, 'Pa', is also computed, representing the XOR operation of all data bits and check bits. Furthermore, an overall check code, 'Ga', is calculated as the XOR operation of 'Pa', all data bits, and

DOI: 10.54254/2755-2721/32/20230186

the inverse of all check bits [9]. This encoding procedure ensures that, even in the event of transmission errors, the original data can be reconstructed using the error detection and correction mechanism.

Upon receiving the extended Hamming code, the receiver first checks the overall code 'Ga' to identify if any error has occurred. Subsequently, based on the parity check bit 'Pa' and the error detection code, the type of error is determined, located, and corrected. This procedure guarantees the data's integrity and accuracy, even under conditions of high error rate inherent in deep space communication [10]. Furthermore, in comparison with other coding techniques, the extended Hamming code more accurately locates and distinguishes errors, particularly in handling multi-bit errors resulting from the complex environment of deep space communication. This provides a more reliable error correction mechanism, conferring a substantial advantage in high-demand applications such as deep space communication. Thus, the use of the extended Hamming code in deep space communication is of paramount importance.

3. System analysis and application research

3.1. Implementation of extended hamming code code in IoT

In the communication of IoT devices, Bluetooth, Wi-Fi, and LoRa are three commonly used protocols, and extended Hamming code plays a crucial role in these protocols.

Bluetooth, as a short-range wireless communication technology, is mainly used for small-scale data transmission between devices. Due to its limited working distance, signals may be interfered with or blocked, leading to data loss or errors. In this case, the extended Hamming code becomes especially important. At the sending end, the original data is converted into a binary data stream, and then according to the encoding rules of the extended Hamming code, corresponding check bits are generated for each data bit, and the overall check bit 'Pa' and the overall check code 'Ga' are calculated to form the extended Hamming code. At the receiving end, the device will perform error detection on the received extended Hamming code, and locate the error bit according to the error detection code, and make the corresponding error correction, thereby ensuring the integrity and accuracy of data even in the event of interference or obstruction during data transmission.

Wi-Fi is a common wireless communication protocol, suitable for high-speed data transmission between devices. Due to the long communication distance and fast data transmission speed of Wi-Fi, data transmission errors may occur. This requires the use of extended Hamming code for encoding and decoding, to effectively detect and correct errors in data, ensuring the reliability of high-speed data transmission. Similarly, Hamming code verification processing is performed at both the sending and receiving ends. This way, even if errors occur in high-speed data transmission, the integrity and accuracy of data can be ensured.

LoRa is a long-range low-power wireless communication technology, mainly used for the construction of wide area networks. Due to its long working distance and low data transmission rate, LoRa is more susceptible to environmental noise, leading to data errors. In this environment, the error detection and correction capability of the extended Hamming code becomes particularly important. At the sending end, the device converts the original data into a binary data stream, and then generates the extended Hamming code according to the encoding rules of the extended Hamming code. At the receiving end, the device decodes the received extended Hamming code. If errors exist, it locates the error bit according to the error detection code and makes corresponding error correction. This way, even if errors occur in long-distance, low-speed data transmission, the integrity and accuracy of data can be ensured.

3.2. IoT device communication

IoT device communication faces numerous challenges, including noise interference, electromagnetic reflections, Doppler effects, and multipath propagation. To address these hurdles, time-varying encryption algorithms and extended Hamming accumulation codes have been introduced. The time-varying encryption algorithm is dynamic and can automatically adjust the encryption strategy according to environmental changes, thereby enhancing communication security. Simultaneously, the extended

DOI: 10.54254/2755-2721/32/20230186

Hamming accumulation code, an effective error correction code, can detect, locate, and correct errors, considerably improving communication reliability. This method is particularly beneficial for audio signal transmission, where it can drastically reduce the bit error rate. Moreover, the inclusion of Irregular Repeat Accumulation (IRA) codes, efficient Forward Error Correction (FEC) codes, provides superior performance to turbo codes while maintaining the same coding complexity..

3.3. IoT device performance

Enhancing IoT device performance requires the adoption of multiple strategies. Artificial Neural Networks (ANN) are utilized for Hamming code decoding due to their potent nonlinear fitting abilities and parallel processing capabilities. This approach significantly accelerates the decoding process and further improves decoding accuracy thanks to ANN's adaptive learning ability. Additionally, extended Hamming accumulation codes (EHA) and nonsystematic irregular repeat accumulation codes (IRA) not only enhance communication reliability but also optimize the device's energy consumption. The performance of IRA codes surpasses turbo codes with the same coding complexity, implying higher communication performance at the same energy cost can be achieved.

3.4. IoT security

To strengthen the security of IoT devices, a time-varying encryption algorithm is employed to tackle dynamically changing threats. This algorithm can adjust the encryption strategy based on environmental changes, thereby improving communication security. Simultaneously, the use of extended Hamming accumulation codes and nonsystematic irregular repeat accumulation codes (IRA) not only enhances communication reliability but also fortifies the system's security. These codes can detect and correct errors during the transmission process, thus preventing malicious attackers from exploiting these errors to attack the system. Furthermore, the use of Artificial Neural Networks for decoding, thanks to ANN's adaptive learning ability, can improve the accuracy of decoding, thereby further enhancing the system's security.

4. Experimental methods

In this study, a series of experiments were designed and conducted to test and validate the performance of extended Hamming codes and artificial neural networks in IoT devices. Initially, a simulation of an IoT communication environment was established, which included devices for simulating potential interferences such as noise, electromagnetic reflection, Doppler effects, and multipath propagation. Within this environment, the impact of integrating extended Hamming codes on the communication performance of IoT devices was observed.

The experimental method for extended Hamming codes mainly includes the following steps: Construction of the generator matrix: Firstly, a generator matrix was built, which is used to multiply the input message bits to get a codeword. The generator matrix is composed of a kxk identity matrix and a kxr parity-check matrix, where k is the number of message bits and r is the number of parity-check bits. Design of the decoder: The input to the decoder might be a received codeword with or without errors. In order to detect and correct errors, a parity-check matrix was utilized. The parity-check matrix is made up of the transpose of the parity-check matrix and an identity matrix. Design of the extended Hamming codes: The proposed extended Hamming code design includes an encoder and a decoder for reliable data transmission. With this extended Hamming code technique, single bit and double adjacent bit errors can be identified and corrected. In the encoding part, the parity-check bits are calculated and appended with the message bits for transmission. At the decoder end, if any errors exist, they are corrected, and the parity-check bits are removed. The uncorrected data is then sent as output. Furthermore, artificial neural networks were also employed for Hamming code decoding, and the performance and efficiency of ANN in the decoding process were studied. The specific experimental methods are as follows: Weight initialization: At the beginning of training, small random values were assigned to the weights.

Calculation of forward responses: Neurons in the input layer receive input patterns. These neurons pass this information to neurons in the hidden layer. The hidden layer neurons calculate the output using

DOI: 10.54254/2755-2721/32/20230186

a nonlinear activation function and the weights from the input to the hidden layer, as well as the input. The output of the hidden layer neurons becomes the input for the output layer neurons. Error backpropagation: The error between the target output signal and the actual output signal was calculated, and for all training inputs, this error should be minimized (possibly zero). Weight update: Based on this error, the weights between the hidden-output layer and the input-hidden layer were updated.

A dedicated testing platform was used, allowing for detailed performance testing of devices under controlled conditions. The platform was utilized to assess the performance changes brought about by the introduction of extended Hamming codes and artificial neural networks, such as data transmission speed, transmission accuracy, and device operation efficiency. The innovation of this method lies in its ability to not only detect and correct single bit errors but also detect and correct double adjacent bit errors. This is extremely important for improving the reliability of data transmission. At the same time, artificial neural networks were used to decode the received data, which is a method of real-time operation, self-organization, and adaptive learning. This method has high efficiency and accuracy when dealing with complex decoding problems.

5. Challenges

Despite promising results obtained from the research, several challenges persist in the practical application of the extended Hamming code. Although theoretically, the extended Hamming code enhances communication reliability, its efficacy could be influenced by factors such as the complexity of the communication environment, performance constraints of the devices, and hardware and software limitations to realize extended Hamming codes.

Furthermore, while artificial neural networks (ANN) have been proven effective for Hamming code decoding, handling of large-scale data and high-speed data transmission may present challenges. The requirement of substantial data and computational resources for ANN training could be problematic in resource-constrained environments.

Lastly, combined use of the extended Hamming code and ANN may give rise to new problems, such as how to effectively integrate these two technologies, and how to balance their impact on system performance and security.

6. Conclusion

This study underscores the immense potential of utilizing extended Hamming codes and Artificial Neural Networks to enhance the communication performance and security of IoT devices. Experimental outcomes suggest that these technologies can substantially augment data transmission speed and precision, elevate the operational efficiency of devices, and bolster device security. However, despite these promising findings, certain obstacles still persist when it comes to real-world applications. These include constraints related to hardware and software when implementing complex extended Hamming codes, and issues associated with managing large-scale data and high-speed data transmission. Consequently, there is an urgent need for future research to delve into how these challenges can be surmounted to maximize the advantages of these technologies. Furthermore, exploring the joint utilization of extended Hamming codes and ANN warrants further investigation. Although each technology brings its own set of benefits to the table, identifying an effective synergy to attain superior performance and heightened security remains an open question. In conclusion, this study introduces fresh perspectives and innovative ideas towards the optimization of IoT devices. It is our hope that these insights will inspire future research to propel advancements in this field and offer practical guidelines for the design and implementation of IoT devices.

References

[1] Xiong L, Han X, Zhong X, et al. RSIS: A secure and reliable secret image sharing system based on extended Hamming codes in industrial Internet of Things[J]. IEEE Internet of Things Journal, 2021, 10(3): 1933-1945.

DOI: 10.54254/2755-2721/32/20230186

[2] Isakov D A, Sokolov A V. McELIECE CRYPTOSYSTEM BASED ON QUATERNARY HAMMING CODES[J]. Informatics & Mathematical Methods in Simulation, 2022, 12(4).
[3] Torres-Alvarado A, Morales-Rosales L A, Algredo-Badillo I, et al. An SHA-3 Hardware Architecture against Failures Based on Hamming Codes and Triple Modular Redundancy[J]. Sensors, 2022, 22(8): 2985.
[4] He Y, Xiao C, Wang S, et al. Smart all-time vision: The battery-free video communication for urban administration and law enforcement[J]. Digital Communications and Networks, 2023.
[5] Cintas-Canto A, Kermani M M, Azarderakhsh R. Error Detection Constructions for ITA Finite Field Inversions Over on FPGA Using CRC and Hamming Codes[J]. IEEE Transactions on Reliability, 2022.
[6] Septien-Hernandez J A, Arellano-Vazquez M, Contreras-Cruz M A, et al. A Comparative study of post-quantum cryptosystems for Internet-of-Things applications[J]. Sensors, 2022, 22(2): 489.
[7] Al Homssi B, Dakic K, Maselli S, et al. IoT network design using open-source LoRa coverage emulator[J]. IEEE access, 2021, 9: 53636-53646.
[8] Nguyen C D, Nguyen P D, Nguyen A T, et al. Performance Evaluation Of Neural Network-Based Channel Detection For STT-MRAM[C]//2021 8th NAFOSTED Conference on Information and Computer Science (NICS). IEEE, 2021: 430-434.
[9] Nguyen T A, Lee J. Improving Bit-Error-Rate Performance Using Modulation Coding Techniques for Spin-Torque Transfer Magnetic Random Access Memory[J]. IEEE Access, 2023, 11: 33005-33013.
[10] Larue G, Dufrene L A, Lampin Q, et al. Neural Belief Propagation Auto-Encoder for Linear Block Code Design[J]. IEEE Transactions on Communications, 2022, 70(11): 7250-7264.