Machine Learning Based Firmware Vulnerability Detection Using Opcode Analysis

C. Siva Prasad, U. Sai Vigneshwaran, R. Senthil Kumar, Dr. S. Subha
Department of Electronics and Communication Engineering, K.L.N. College of Engineering, Pottapalayam, Sivaganga district.

siva.prasad0220@gmail.com, saivigneshwaran2005@gmail.com, senthilkumar200399@gmail.com, subamanian14@gmail.com

How to Cite this Article:

Prasad, C. S., Vigneshwaran, U. S., Kumar, R. S. & Subha, S. (2026). Machine Learning Based Firmware Vulnerability Detection Using Opcode Analysis. International Journal of Creative and Open Research in Engineering and Management, 02(03). https://doi.org/10.55041/ijcope.v2i3.251

https://doi.org/10.55041/ijcope.v2i3.251

Abstract --- *The rapid expansion of Internet of Things (IoT) devices across smart homes, healthcare, agriculture, and industrial systems has increased the importance of firmware security. Firmware acts as the operational core of IoT devices, and vulnerabilities within it can lead to unauthorized access, data leakage, device malfunction, and large-scale cyberattacks. This study presents an IoT Firmware Vulnerability Detection System that identifies security risks in firmware files using a hybrid approach that combines machine learning with analytical feature-based assessment. The proposed system accepts firmware in BIN, HEX, and ELF formats and processes them through a structured workflow that includes file normalization, architecture detection, binary feature extraction, opcode analysis, and classification. Key features such as entropy, opcode diversity, string count, byte variance, and binary size are extracted to represent the behavioral characteristics of the firmware. These features are then analyzed using a Random Forest model to estimate vulnerability probability. To improve reliability and interpretability, the machine learning output is combined with heuristic risk scoring to classify firmware as safe, suspicious, or vulnerable. The results show that the hybrid framework provides effective and understandable vulnerability assessment while supporting multiple firmware types and architectures. The system also includes a user-oriented interface that presents risk scores, feature-based visualizations, and backend analysis logs for better transparency. This research is significant because it offers a practical, scalable, and explainable solution for strengthening IoT firmware security, supporting researchers, developers, and cybersecurity professionals in the early detection of firmware-level threats.*

Keywords --- firmware analysis, iot security, machine learning, opcode analysis, vulnerability detection

I. INTRODUCTION

The rapid growth of Internet of Things (IoT) devices in applications such as smart homes, healthcare, and industrial automation has significantly increased security concerns. Firmware, which serves as the core software of embedded systems, is a critical component that is often targeted by attackers.

Analyzing firmware for vulnerabilities is challenging due to the lack of access to source code and the diversity of hardware architectures used in IoT devices. Traditional security analysis methods, including static and dynamic analysis, require extensive expertise and are time-consuming. Moreover, many existing approaches depend on source code availability, which is not feasible for proprietary firmware.

To address these challenges, this paper proposes an automated Machine Learning based approach for detecting vulnerabilities in firmware using opcode sequence analysis. The proposed system directly analyzes binary firmware by converting it into opcode sequences through disassembly. These sequences are processed using machine learning techniques, where a transformer-based model extracts meaningful features and a Random Forest classifier identifies whether the firmware is safe or vulnerable.

The main contributions of this work are as follows:

- A source code-independent firmware analysis approach
- An automated pipeline for opcode extraction and processing
- A machine learning-based classification model for vulnerability detection
- Support for multiple microcontroller architectures such as Arduino, ESP32, and STM32

This approach provides a practical and scalable solution for enhancing IoT firmware security.

II. PROPOSED METHODOLOGY

A. SYSTEM OVERVIEW

The proposed system follows a structured pipeline:

1. Firmware Input (HEX/BIN file)
2. Conversion to Binary Format
3. Disassembly
4. Opcode Extraction
5. Tokenization
6. Feature Extraction using Transformer Model
7. Classification using Random Forest

B. FIRMWARE PREPROCESSING

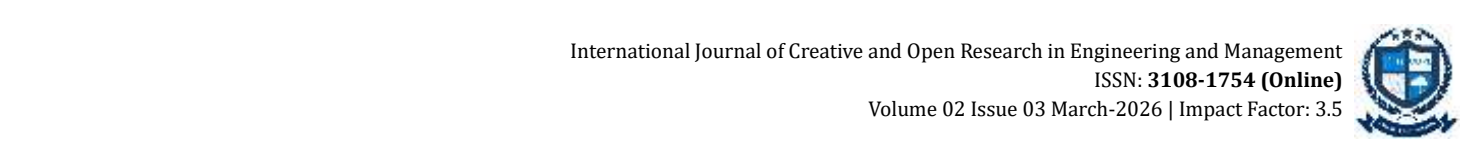
The input firmware file is first converted into binary format if required. Tools such as disassemblers are used to convert binary files into assembly code, from which opcode sequences are extracted.

C. OPCODE SEQUENCE PROCESSING

Opcode sequences are tokenized to convert them into machine-readable format. These sequences represent the behavior of the firmware and serve as input for the machine learning model.

D. FEATURE EXTRACTION

A machine learning based approach is used to analyze opcode sequences for vulnerability detection. The extracted opcode sequences are converted into numerical feature representations using suitable encoding techniques.



E. CLASSIFICATION

A Random Forest classifier is used to categorize firmware into:

- SAFE
- VULNERABLE

This model is chosen due to its robustness and ability to handle complex data patterns.

III. SYSTEM ARCHITECTURE

The overall architecture of the proposed system is illustrated in Fig. 1. The system processes firmware through multiple stages including binary conversion, disassembly, opcode extraction, feature encoding, and classification.

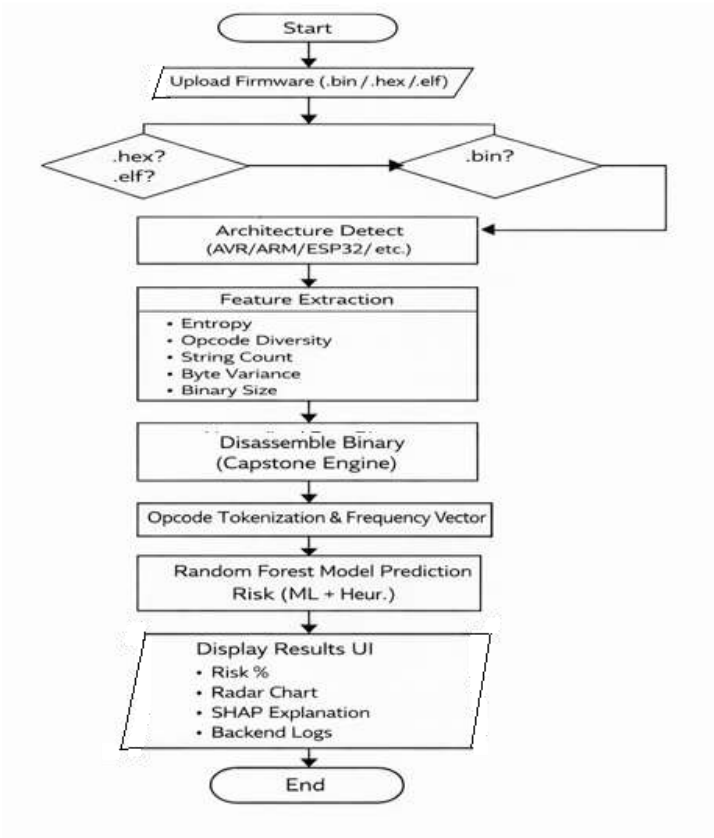


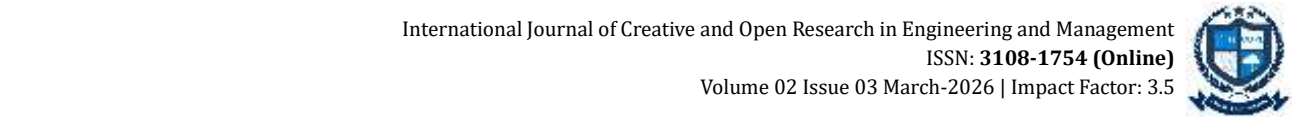
Fig. 1. Proposed System Architecture

The process begins with firmware input, which is converted into binary format. The binary is then disassembled to extract opcode sequences. These opcodes are transformed into numerical features using encoding techniques. Finally, a Random Forest classifier analyzes the features and predicts whether the firmware is safe or vulnerable.

The system is designed to support multiple microcontroller architectures, including:

- Arduino (AVR)
- ESP32
- STM32

Each firmware is processed and stored in architecture-specific folders. The backend automatically performs all operations, ensuring ease of use.



IV. Results and Discussion

The proposed system was tested using sample firmware datasets. The results indicate:

- High accuracy in vulnerability detection
- Efficient processing of binary-only firmware
- Scalability across multiple architectures

The system demonstrates significant improvement over traditional manual analysis methods.

The proposed IoT firmware vulnerability detection system was tested using multiple firmware samples. The system analyzes firmware using opcode-based feature extraction and a Random Forest classifier to determine whether the firmware is safe or vulnerable. The results obtained demonstrate the effectiveness of the proposed approach.

Fig. 2. Output of Firmware Vulnerability Detection System

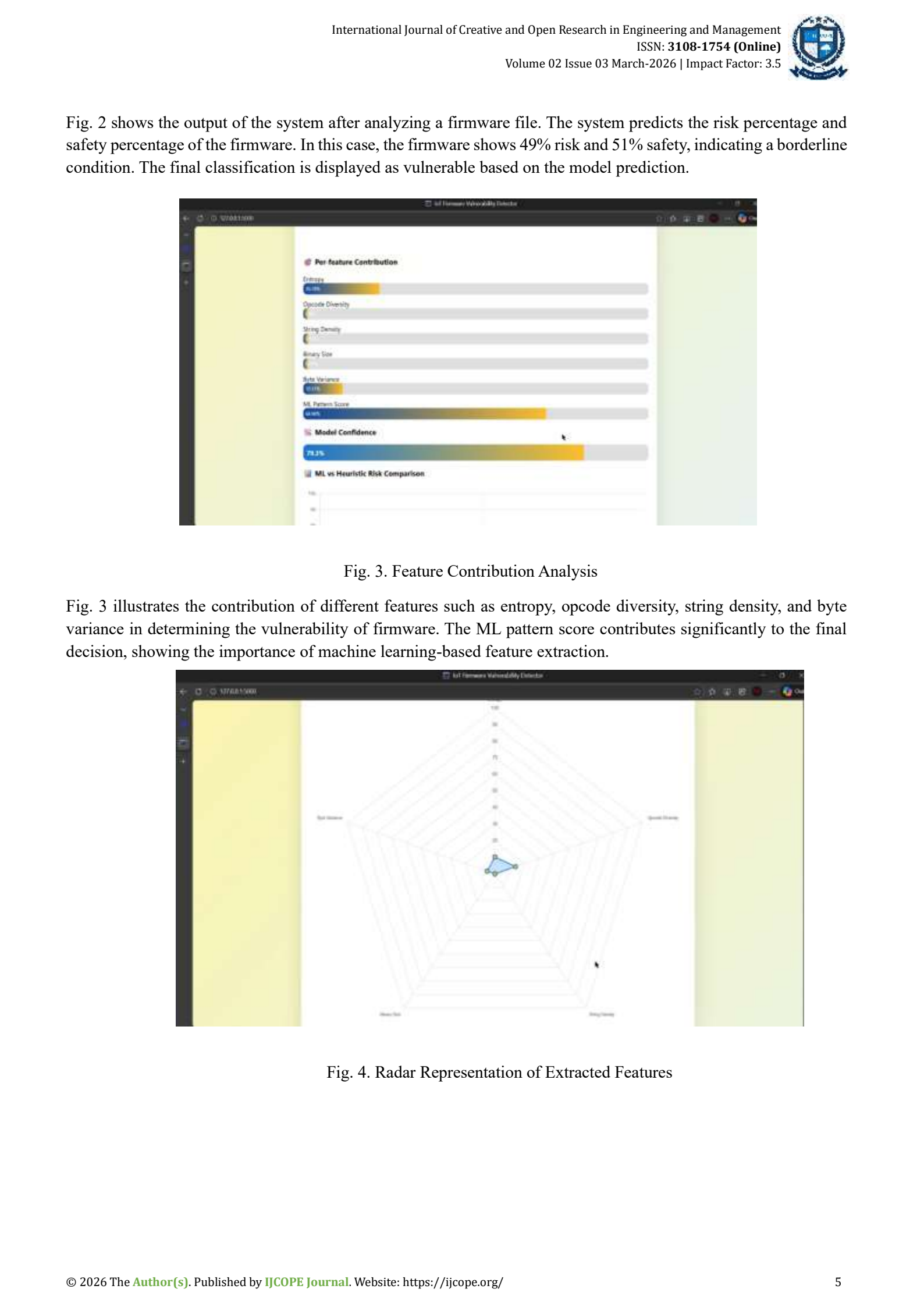


Fig. 2 shows the output of the system after analyzing a firmware file. The system predicts the risk percentage and safety percentage of the firmware. In this case, the firmware shows 49% risk and 51% safety, indicating a borderline condition. The final classification is displayed as vulnerable based on the model prediction.

Fig. 3. Feature Contribution Analysis

Fig. 3 illustrates the contribution of different features such as entropy, opcode diversity, string density, and byte variance in determining the vulnerability of firmware. The ML pattern score contributes significantly to the final decision, showing the importance of machine learning-based feature extraction.

Fig. 4. Radar Representation of Extracted Features

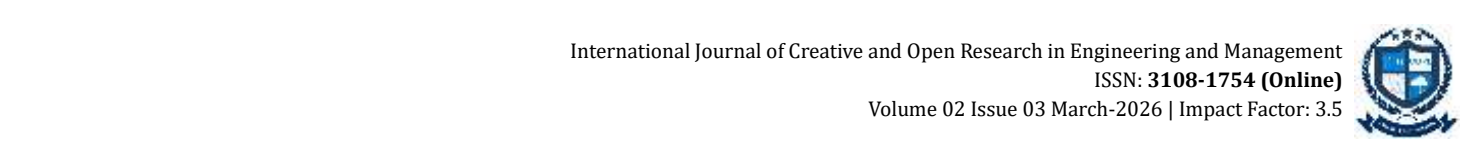


Fig. 4 represents the extracted features in the form of a radar chart. This visualization helps in understanding the distribution of various features such as entropy, opcode diversity, and binary size, which are used for classification.

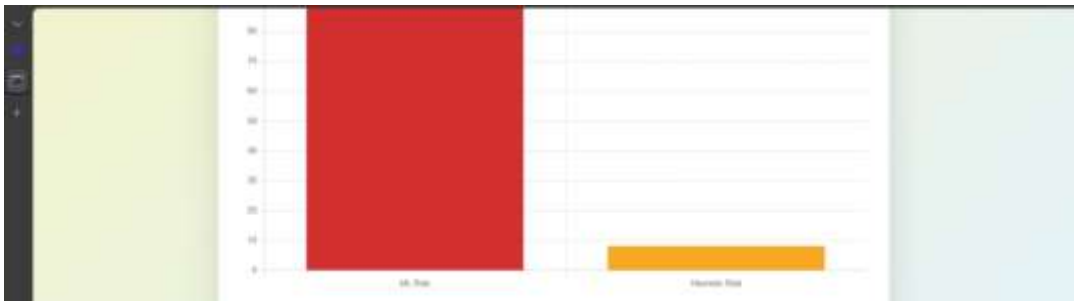


Fig. 5. Comparison of ML-Based and Heuristic Risk Analysis

Fig. 5 compares the risk prediction obtained from the machine learning model and heuristic methods. The ML-based approach provides more accurate and reliable predictions compared to traditional heuristic techniques.

Fig. 6. Backend Processing and Opcode Extraction Steps

Fig. 6 shows the backend processing steps involved in firmware analysis, including format detection, disassembly, opcode extraction, tokenization, and feature generation. This demonstrates the automated pipeline implemented in the system.

V. Applications

The proposed system can be used in:

- IoT device security auditing
- Embedded system development
- Cybersecurity research
- Firmware validation in industries

VI. Conclusion

This paper presents an Machine Learning based approach for detecting vulnerabilities in IoT firmware using opcode sequences. The system eliminates the need for source code and provides an automated solution for firmware analysis. Future work includes improving model accuracy using deep learning techniques and expanding support for more architectures.

REFERENCES

1. Kumar, S. Jain, and R. Gupta, "AI-Driven Vulnerability Detection in IoT Firmware Using Opcode Sequence Embeddings," IEEE Transactions on Dependable and Secure Computing, vol. 22, no. 1, pp. 45–58, Jan. 2025.
2. L. Chen and M. Zhao, "Machine Learning Techniques for Binary Firmware Security Analysis," *IEEE Access*, vol. 13, pp. 12134–12147, 2025.
3. Y. Singh and T. Patel, "Exploring Cross-Architecture Opcode Patterns for Embedded Firmware Vulnerability Detection," *2025 IEEE International Conference on Cybersecurity and Resilience (CCR)*, pp. 98–106, Mar. 2025.
4. J. Park and H. Lee, "A Random Forest Based Framework for Detection of Security Vulnerabilities in Embedded Firmware," *IEEE Internet of Things Journal*, vol. 12, no. 5, pp. 5578–5589, May 2025.