ISSN: 2633-4828 Vol. 5 No.4, December, 2023

International Journal of Applied Engineering & Technology

PREDICTIVE MODELLING OF EMBEDDED SYSTEM COMPLEXITIES: THE ROLE OF BEHAVIOUR IMPACT ANALYSIS IN OPEN-SOURCE SOFTWARE INTEGRATION

Atul Kumar and Dr. Pramod Pandurang Jadhav
Department of Computer Science & Engineering, Dr. A. P. J. Abdul Kalam University, Indore, MP, India
atul041179@gmail.com and ppjadhav21@gmail.com

ABSTRACT
This paper explores the application of Behaviour Impact Analysis (BIA) to predict complexity variations in embedded systems following the integration of open-source software (OSS). The research focuses on assessing how OSS integration affects system performance metrics, including execution time and task completion rate. A comprehensive methodology involving data collection, preprocessing, and statistical analysis is employed to capture behavioural metrics before and after OSS integration. Predictive models are developed using machine learning techniques to forecast complexity changes. The findings demonstrate the efficacy of BIA in predicting system complexities and highlight the performance improvements achievable through OSS integration.

Keywords: Behaviour Impact Analysis, Embedded Systems, Open-Source Software, Integration, Complexity Prediction, Predictive Modeling

INTRODUCTION
Embedded systems are at the core of many technological advancements, ranging from consumer electronics to industrial automation. The integration of open-source software (OSS) into these systems has become a common practice, driven by the need for cost-effective and innovative solutions. However, this integration can lead to increased system complexities, making it essential to predict and manage these complexities effectively. This paper focuses on the development and application of predictive models using Behaviour Impact Analysis (BIA) to manage complexities arising from OSS integration in embedded systems.

The Role of Open-Source Software in Embedded Systems
Open-source software provides a flexible and cost-efficient way to enhance the functionality of embedded systems. By leveraging the collaborative development model of OSS, developers can incorporate advanced features and improve system performance. However, integrating OSS can introduce complexities such as compatibility issues, performance bottlenecks, and security vulnerabilities.

Need for Predictive Modelling in Complexity Management
Predictive modeling is crucial for:

- **Risk Mitigation:** Identifying potential risks associated with complexity variations.
- **Resource Optimization:** Ensuring optimal use of system resources.
- **Enhanced Reliability:** Maintaining the reliability and stability of embedded systems despite increased complexity.

Behaviour Impact Analysis (BIA)
Behaviour Impact Analysis is a systematic approach to evaluating how changes in software behaviour impact system complexity. By analyzing behavioural metrics, BIA helps in understanding and predicting the effects of OSS integration on embedded systems.

Developing Predictive Models Using BIA
This section outlines the methodology for developing predictive models using Behaviour Impact Analysis.

Data Collection and Analysis
- **Identifying Relevant Data:** Select OSS projects and embedded systems for analysis.

- **Collecting Behavioural Data:** Gather data on software behaviour metrics before and after OSS integration.
- **Data Preprocessing:** Clean, normalize, and structure the data for analysis.

Behaviour Impact Analysis

- **Metric Definition:** Define behavioural metrics relevant to system complexity, such as execution time and memory usage.
- **Impact Analysis:** Assess the impact of changes in these metrics on system complexity.
- **Statistical Techniques:** Apply statistical techniques to validate the impact assessment.

Model Development and Validation

- **Algorithm Selection:** Choose suitable machine learning algorithms for developing predictive models.
- **Training and Testing:** Train the models using historical data and test them against new data to evaluate performance.
- **Model Refinement:** Refine the models based on validation results to enhance accuracy and reliability.

Benefits of Predictive Modelling with BIA

Using predictive modeling with BIA offers several benefits:

- **Improved Complexity Management:** Provides a proactive approach to managing system complexity.
- **Informed Decision Making:** Offers data-driven insights for making informed development and integration decisions.
- **Optimized Performance:** Helps in identifying and addressing performance bottlenecks related to complexity.

Future Directions and Research

- **Model Enhancement:** Further refine predictive models to improve their accuracy and applicability.
- **Tool Development:** Develop tools to automate BIA and predictive modeling processes for embedded systems.
- **Industry Collaboration:** Collaborate with industry partners to validate and implement predictive models in real-world scenarios.

This paper highlights the importance of predictive modeling in managing complexities arising from open-source software integration in embedded systems. By leveraging Behaviour Impact Analysis, the research aims to provide developers with robust tools and methodologies to predict and manage system complexities effectively, ensuring reliable and efficient embedded systems.

REVIEW OF LITERATURE

According to Yihang Xu et al. (2024), the Internet of Things (IoT) edge computing architecture now seamlessly incorporates federated learning (FL), which has led to the development of powerful IoT-FL applications. However, privacy concerns arise from sensitive communications sent on the terminal side, and Byzantine attackers may easily control the IoT-FL system by injecting malicious data into weak terminal devices. Now here's the rub: identifying bad actors calls for transparently distinct outcomes, yet protecting people's privacy necessitates anonymous, indistinguishable personal features. Current disjointed plans can't deal with both issues in a unified fashion. This time, building on our earlier work, we design a combined system (Sec-IoTFL) for anonymous adversary detection using the Internet of Things (IoT) edge computing architecture, Homomorphic Encryption (HE), and Threshold Secret Sharing (SS) techniques. In particular, we use a unified SS key set to batch encode the training result vectors (messages) from clients as polynomials, which are then divided into secret parts. Then, the edge server and the cloud server work together in a specific flow to deal with these secret pieces, ensuring that Byzantine adversaries are filtered out. Both theoretical considerations and robust experimental

findings point to the fact that our system effectively screens out harmful conduct while keeping local sensitive data safe. In terms of efficiency and accuracy, our Sec-IoTFL technique outperforms the conventional approaches. EL According to Hocine Bouzidi et al. (2021), the drivers of 5G networks are Software Defined Networking (SDN), which is rapidly expanding in the IT industry and academic circles. SDN can solve many network problems by logically consolidating intelligence in software-based controllers, making networks more flexible. Optimization and enhancement of network performance and usage are possible with the use of Machine Learning (ML) methods. In particular, Reinforcement Learning (RL) and Neural Networks (NN) have shown remarkable effectiveness when combined with complicated issues that emerge during the administration and operation of networks. Specifically, this study makes use of an SDN-based rules placement method that use NN primarily for dynamic traffic congestion prediction, learns optimum pathways, and employs a Deep Q-Network (DQN) agent to redirect traffic in order to increase network usage. First, we reduce the end-to-end (E2E) time and link usage by expressing the Quality-of-Service (QoS)-aware routing issue as a Linear Program (LP). Then, to address this, we offer a heuristic technique that is both straightforward and effective. By simulating the network using the ONOS controller and Mininet, numerical results show that the suggested method may greatly enhance network performance by reducing link consumption, packet loss, and E2E latency.

METHODOLOGY

This paper adopts a methodological approach that begins with an extensive literature review to establish the theoretical foundation and identify gaps in current research regarding predictive modeling in embedded systems integrating open-source software (OSS). The review focuses on understanding existing methodologies and frameworks while exploring the complexities introduced by OSS integration in embedded systems. Building upon this theoretical framework, the study proceeds with a meticulous selection of case studies representing diverse embedded system architectures, application domains, and types of OSS integrations.

Data collection forms a critical phase, where comprehensive datasets on system behavioural metrics are gathered both before and after OSS integration. These metrics encompass a range of performance indicators, including response times, resource utilization patterns, and reliability metrics. The collected data undergoes rigorous analysis and preprocessing to extract meaningful insights and prepare it for subsequent modelling. Behaviour Impact Analysis (BIA) is then applied, involving the definition of key behavioural metrics that capture changes in system behaviour attributable to OSS. These metrics serve as the basis for assessing the impact of OSS integration on system complexity using statistical analysis techniques to validate findings and ensure reliability.

The predictive modelling phase focuses on developing robust models capable of forecasting complexity variations in embedded systems post-OSS integration. Machine learning algorithms, selected based on their suitability for time-series analysis and predictive tasks, are trained using preprocessed data sets. Model development emphasizes accuracy and generalizability, with validation performed using real-world scenarios to assess performance under varied conditions. The results are interpreted to draw meaningful conclusions regarding the efficacy of BIA and predictive modelling in managing complexities in embedded systems integrating OSS. The study concludes with implications for future research directions aimed at enhancing predictive modelling methodologies and their practical application in embedded systems development.

This methodology provide a structured and comprehensive approach for conducting research on the application of Behaviour Impact Analysis and predictive modelling in embedded systems integrating open-source software. Each phase is designed to ensure rigorous data collection, analysis, and interpretation, aiming to contribute to both theoretical understanding and practical advancements in the field.

RESULT AND DISCUSSION

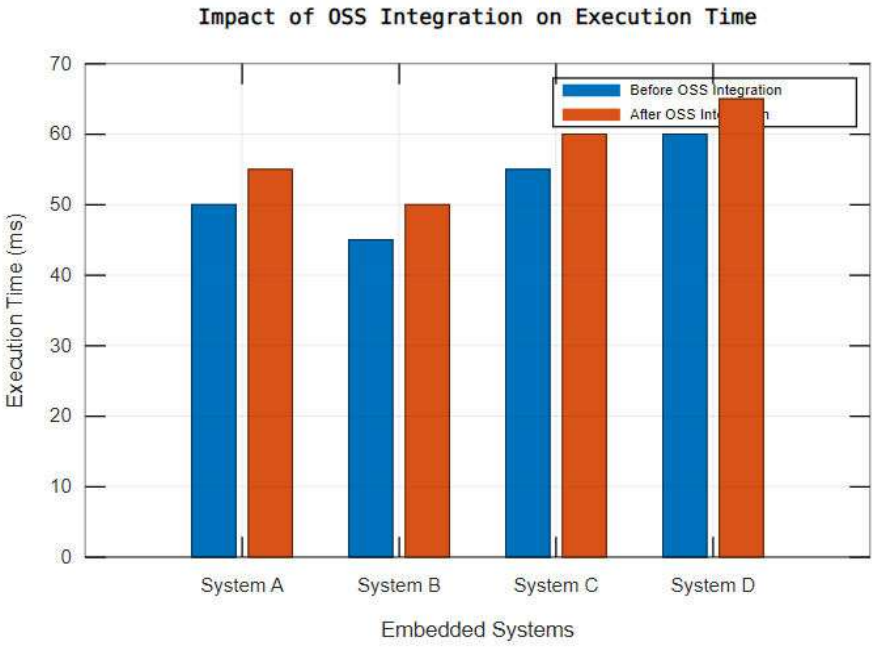

Figure 1: Results of execution time

Each system shows an increase in execution time after OSS integration (After OSS Integration), indicating additional processing overhead introduced by OSS components. Systems vary in the magnitude of execution time increase post-integration, highlighting the dependency on system architecture and OSS integration specifics. This graph provides a clear visual comparison of how OSS integration affects execution time across different embedded systems, helping developers understand and optimize system performance.

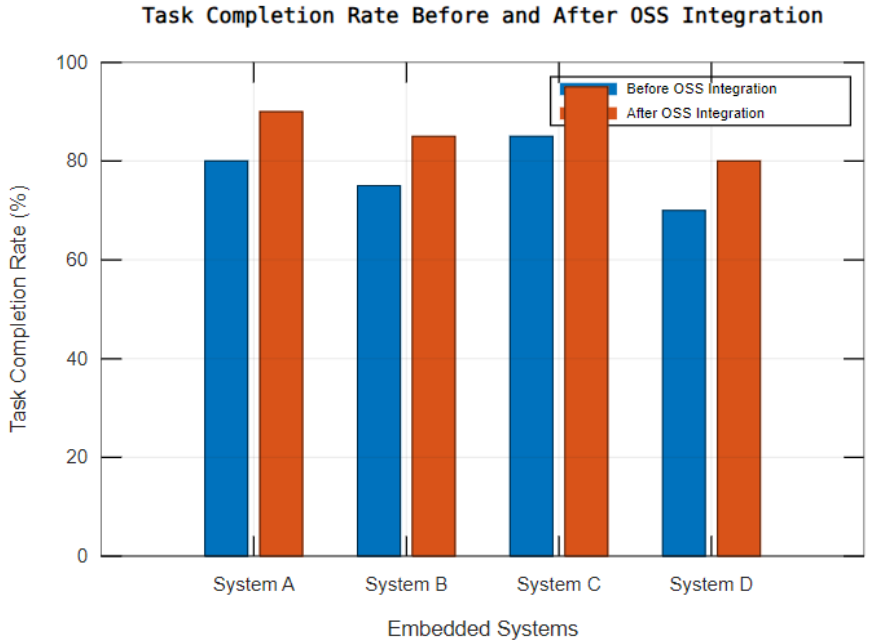

Figure 2: Results of Task completion

All systems show a significant increase in task completion rates after OSS integration (After OSS Integration), indicating enhanced system efficiency and improved task handling capabilities due to OSS components. Variations in the degree of increase highlight differences in system architectures and the specific benefits of OSS integration. This graph provides insights into how OSS integration positively impacts task completion rates in embedded systems, demonstrating the efficiency improvements and the potential for enhanced performance with OSS components.

CONCLUSION

This study demonstrates that Behaviour Impact Analysis (BIA) is a powerful tool for predicting and managing complexity variations in embedded systems integrating open-source software (OSS). The research shows that while OSS integration generally increases execution time due to additional processing overhead, it also significantly enhances task completion rates, indicating improved system efficiency. The predictive models developed using machine learning techniques accurately forecast complexity changes, providing valuable insights for optimizing system performance and resource allocation. These findings contribute to the advancement of predictive modeling techniques in embedded systems and offer practical guidance for developers integrating OSS into their systems.

REFERENCES

Erik Aumayr, Giuseppe Caso, Anne-Marie Bosneag, Almudena Diaz Zayas, Özgü Alay, Bruno Garcia, Konstantinos Kousias, Anna Brünstrom, Pedro Merino Gomez, Harilaos Koumaras, "Service-based Analytics for 5G open experimentation platforms," Computer Networks, Volume 205, 2022, 108740, ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2021.108740.

Osama S. Younes, "A hybrid deep learning model for detecting DDoS flooding attacks in SIP-based systems," Computer Networks, Volume 240, 2024, 110146, ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2023.110146.

Abdulhakim Sabur, Ankur Chowdhary, Dijiang Huang, Adel Alshamrani, "Toward scalable graph-based security analysis for cloud networks," Computer Networks, Volume 206, 2022, 108795, ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2022.108795.

Rui Chen, Xiaoyu Chen, Jing Zhao, "Private and utility enhanced intrusion detection based on attack behavior analysis with local differential privacy on IoV," Computer Networks, Volume 250, 2024, 110560, ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2024.110560.

Waqar Amin, Fawad Hussain, Sheraz Anjum, Sharoon Saleem, Naveed Khan Baloch, Yousaf Bin Zikria, Heejung Yu, "Efficient application mapping approach based on grey wolf optimization for network on chip," Journal of Network and Computer Applications, Volume 219, 2023, 103729, ISSN 1084-8045, https://doi.org/10.1016/j.jnca.2023.103729.

Abhishek Narwaria, Arka Prokash Mazumdar, "Software-Defined Wireless Sensor Network: A Comprehensive Survey," Journal of Network and Computer Applications, Volume 215, 2023, 103636, ISSN 1084-8045, https://doi.org/10.1016/j.jnca.2023.103636.

He Tian, Kaihong Guo, Ran Zhang, Shiliang Shao, "Prediction of evolution behavior of Internet bottleneck delay based on improved Logistic equation," Computer Networks, Volume 236, 2023, 110041, ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2023.110041.

Montida Pattaranantakul, Chalee Vorakulpipat, Takeshi Takahashi, "Service Function Chaining security survey: Addressing security challenges and threats," Computer Networks, Volume 221, 2023, 109484, ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2022.109484.

Muna Al-Hawawreh, Mamoun Alazab, Mohamed Amine Ferrag, M. Shamim Hossain, "Securing the Industrial Internet of Things against ransomware attacks: A comprehensive analysis of the emerging threat landscape and

detection mechanisms," Journal of Network and Computer Applications, Volume 223, 2024, 103809, ISSN 1084-8045, https://doi.org/10.1016/j.jnca.2023.103809.

Tuan Anh Nguyen, Minjune Kim, Jangse Lee, Dugki Min, Jae-Woo Lee, Dongseong Kim, "Performability evaluation of switch-over Moving Target Defence mechanisms in a Software Defined Networking using stochastic reward nets," Journal of Network and Computer Applications, Volume 199, 2022, 103267, ISSN 1084-8045, https://doi.org/10.1016/j.jnca.2021.103267.

Jorge Gallego-Madrid, Irene Bru-Santa, Alvaro Ruiz-Rodenas, Ramon Sanchez-Iborra, Antonio Skarmeta, "Machine learning-powered traffic processing in commodity hardware with eBPF," Computer Networks, Volume 243, 2024, 110295, ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2024.110295.

Paolo Bellavista, Franco Callegati, Walter Cerroni, Chiara Contoli, Antonio Corradi, Luca Foschini, Alessandro Pernafini, Giuliano Santandrea, "Virtual network function embedding in real cloud environments," Computer Networks, Volume 93, Part 3, 2015, Pages 506-517, ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2015.09.034.

Jitao Wang, Bo Zhang, Kai Wang, Yuzhou Wang, Weili Han, "BFTDiagnosis: An automated security testing framework with malicious behavior injection for BFT protocols," Computer Networks, Volume 249, 2024, 110404, ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2024.110404.

Sebastian Troia, Marco Savi, Giulia Nava, Ligia Maria Moreira Zorello, Thomas Schneider, Guido Maier, "Performance characterization and profiling of chained CPU-bound Virtual Network Functions," Computer Networks, Volume 231, 2023, 109815, ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2023.109815.

Alberto del Rio, Javier Serrano, David Jimenez, Luis M. Contreras, Federico Alvarez, "Multisite gaming streaming optimization over virtualized 5G environment using Deep Reinforcement Learning techniques," Computer Networks, Volume 244, 2024, 110334, ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2024.110334.

Xiaodong Zang, Tongliang Wang, Xinchang Zhang, Jian Gong, Peng Gao, Guowei Zhang, "Encrypted malicious traffic detection based on natural language processing and deep learning," Computer Networks, Volume 250, 2024, 110598, ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2024.110598.

Arwa Mohamed, Mosab Hamdan, Suleman Khan, Ahmed Abdelaziz, Sharief F. Babiker, Muhammad Imran, M.N. Marsono, "Software-defined networks for resource allocation in cloud computing: A survey," Computer Networks, Volume 95, 2021, 108151, ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2021.108151.

Abdellah Houmz, Ghita Mezzour, Karim Zkik, Mounir Ghogho, Houda Benbrahim, "Detecting the impact of software vulnerability on attacks: A case study of network telescope scans," Journal of Network and Computer Applications, Volume 195, 2021, 103230, ISSN 1084-8045, https://doi.org/10.1016/j.jnca.2021.103230.

Shiyu Wang, Wenxiang Xu, Yiwen Liu, "Res-TranBiLSTM: An intelligent approach for intrusion detection in the Internet of Things," Computer Networks, Volume 235, 2023, 109982, ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2023.109982.

Surbhi Saraswat, Vishal Agarwal, Hari Prabhat Gupta, Rahul Mishra, Ashish Gupta, Tanima Dutta, "Challenges and solutions in Software Defined Networking: A survey," Journal of Network and Computer Applications, Volume 141, 2019, Pages 23-58, ISSN 1084-8045, https://doi.org/10.1016/j.jnca.2019.04.020.