Hybrid DeBERTa-BiLSTM-CNN for Enhanced Smart Contract Vulnerability Detection

Muhammad Sannan Khaliq [1], Md Tayeb Adnan [1], Subroto Kumar Ghosh [1], Love Allen Chijioke Ahakonye [2], Jae Min Lee [1], Dong-Seong Kim [1] *

[1] IT-Convergence Engineering, *Kumoh National Institute of Technology*, Gumi, South Korea

* NSLab Co. Ltd., Gumi, South Korea, *Kumoh National Institute of Technology*, Gumi, South Korea

[2] ICT Convergence Research Center, *Kumoh National Institute of Technology*, Gumi, South Korea

(sannan, mdtayebadnan, subroto, loveahakonye, ljmpaul, dskim)@kumoh.ac.kr

***Abstract*—This study presents a hybrid model combining DeBERTa, BiLSTM, and CNN to detect smart contract vulnerabilities. Traditional detection mechanisms struggle with accuracy and scalability, while deep learning models often face issues with information loss and a lack of labeled datasets. The approach utilizes DeBERTa for contextual word embeddings, BiLSTM for capturing sequential dependencies, and CNN for hierarchical feature extraction. It achieves 0.91 accuracy and a weighted F1-score of 0.91, offering a robust solution for enhancing Ethereum contract security through automated detection and early threat identification.**

***Index Terms*—BiLSTM, BERT, CNN, Smart Contract, vulnerability detection**

I. Introduction

Blockchain, driven by Bitcoin, provides a decentralized and secure framework [1], [2]. Central to its utility are smart contracts, autonomous programs that remove intermediaries [3] and are critical in domains such as DeFi, supply chains, and healthcare. Yet, their vulnerabilities pose serious security risks [1], [4], [5], as highlighted by the DAO attack that resulted in a loss of 3.6 million ETH (about $50 million) [5], [6]. Based on rigid rules, traditional static analysis struggles with precision, scalability, and dynamic behavior capture [1]. Deep learning approaches have emerged to address this, mitigating reliance on manual feature design [1], [5]. However, many still rely on monothetic features or face constraints from limited labeled data, reducing their generalizability [5].

Studies show the significance of hybrid deep learning approaches [7] in vulnerability detection. Hence, this study proposes a hybrid deep learning architecture comprising DeBERTa-BiLSTM-CNN for detecting vulnerabilities in Ethereum smart contracts. DeBERTa's contextualized representations, BiLSTM's sequential modeling, and CNN's pattern extraction capabilities improve the model's detection accuracy, reduce false positives, and enhance scalability. Evaluated on a real-world dataset, the model outperforms existing approaches while addressing key challenges such as limited labeled data and the complexity of smart contract interactions.

II. Proposed System

Figure 1 illustrates the Hybrid DeBERTa-BiLSTM-CNN model for detecting vulnerabilities in Ethereum smart contracts. By combining DeBERTa, BiLSTM, and CNN, the model improves accuracy, reduces false positives, and enhances scalability through contextual, sequential, and hierarchical feature extraction. The model is trained on a dataset of Ethereum contract snippets labeled with vulnerabilities such as Data Consistency (DC), Input/Output (IO), Reentrancy (RE), and Time Dependency (TD) [8]. These labels, provided by domain experts, are based on known attack patterns. The data is preprocessed by removing irrelevant elements, tokenized using the `Sentencepiece` subword, and padded to a fixed length of 3000 tokens. The tokenized sequences are standardized and normalized to the [0, 1] range to improve numerical stability and training efficiency.

The CNN-BiLSTM model integrates convolutional layers for feature extraction and dimensionality reduction with bidirectional LSTMs to capture contextual dependencies in Ethereum smart contract opcode sequences. CNNs identify local vulnerability patterns, while BiLSTMs model forward and backward interactions, which are crucial for detecting vulnerabilities tied to execution flow. A prediction threshold enables classification of known and novel vulnerabilities, making this hybrid architecture a robust solution for smart contract security.

III. Performance Evaluation

The model achieved 91% accuracy, showing effective learning over time. The model learning process curve in Figure 2 illustrates the (a) decreasing loss and (b) increasing accuracy. Table I shows the model performs strongly in RE detection, achieving high precision, recall, and F1-score. IO results are similarly robust, while DC exhibits reduced recall. Despite this, the model maintains a macro-average F1-score of 0.84 and a weighted average of 0.91, indicating effective vulnerability detection with room for improvement in DC recall.

TABLE I: Results

Type of Vulnerabilities	Precision	Recall	F1-Score
DC	0.99	0.5	0.67
IO	0.85	0.90	0.88
RE	0.94	0.97	0.95
TD	0.90	0.84	0.87

IV. Conclusion

This paper presented a Hybrid DeBERTa-BiLSTM-CNN model for smart contract vulnerability detection, achieving

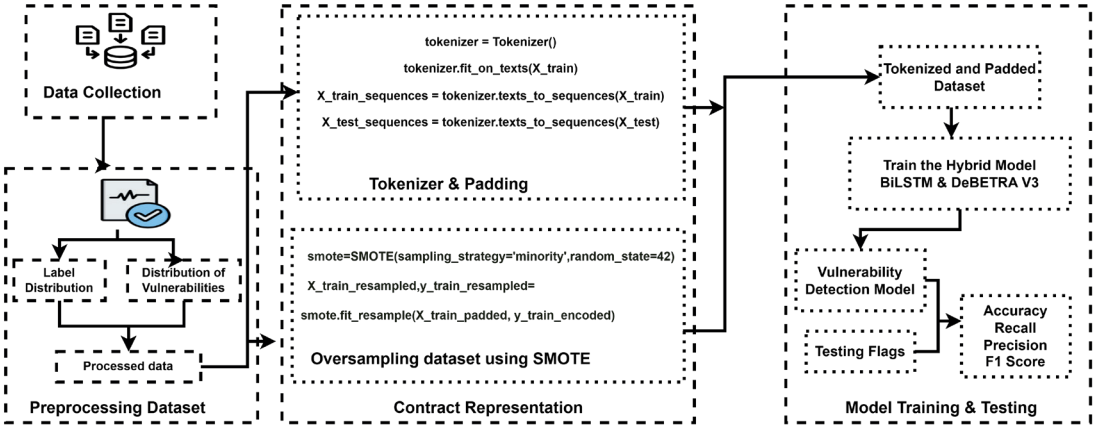

Fig. 1: Proposed Smart Contract Vulnerability Detection System

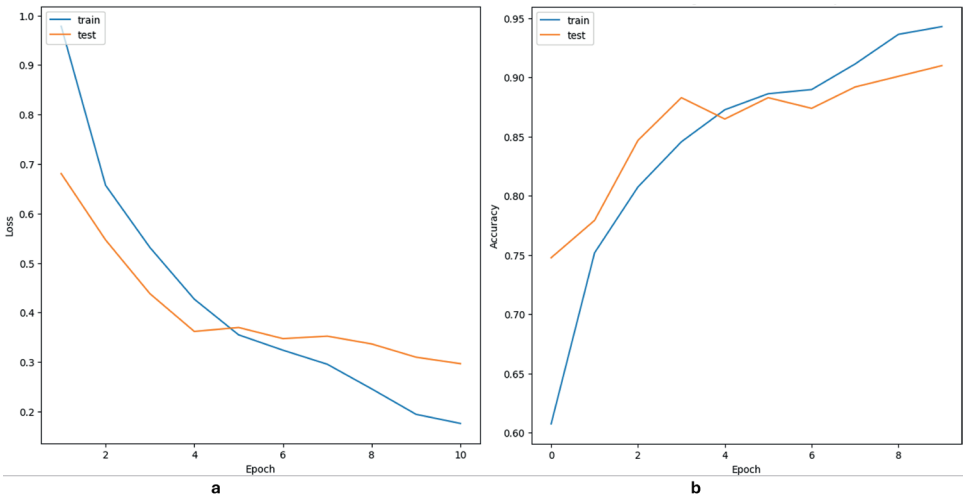

Fig. 2: Learning process path curve of the Hybrid DeBERTa-BiLSTM-CNN Model for detecting Smart Contract Vulnerability

an accuracy of 91%. The model effectively identified RE and IO vulnerabilities, while showing room for improvement in DC vulnerability detection, particularly in recall. These results demonstrate the potential of deep learning techniques to enhance smart contract security by automating and scaling vulnerability detection. Future work will focus on optimizing recall for DC vulnerabilities and expanding the dataset for improved generalization.

Acknowledgment

This work was partly supported by Innovative Human Resource Development for Local Intellectualization program through the IITP grant funded by the Korea government(MSIT) (IITP-2025-RS-2020-II201612, 33%) and by Priority Research Centers Program through the NRF funded by the MEST(2018R1A6A1A03024003, 33%) and by the MSIT, Korea, under the ITRC support program(IITP-2025-RS-2024-00438430, 34%)

References

[1] H. Ding, Y. Liu, X. Piao, H. Song, and Z. Ji, "SmartGuard: An LLM-Enhanced Framework for Smart Contract Vulnerability Detection," *Expert Systems with Applications*, vol. 269, p. 126479, 2025.

[2] L. A. C. Ahakonye, C. I. Nwakanma, and D.-S. Kim, "Tides of Blockchain in IoT Cybersecurity," *Sensors*, vol. 24, no. 10, p. 3111, 2024.

[3] W. Zou, D. Lo, P. S. Kochhar, X.-B. D. Le, X. Xia, Y. Feng, Z. Chen, and B. Xu, "Smart Contract Development: Challenges and Opportunities," *IEEE Transactions on Software Engineering*, vol. 47, no. 10, pp. 2084–2106, 2021.

[4] Y. Lu, "The Blockchain: State-of-The-Art and Research Challenges," *Journal of Industrial Information Integration*, vol. 15, pp. 80–90, 2019.

[5] H. Chu, P. Zhang, H. Dong, Y. Xiao, and S. Ji, "DeepFusion: Smart Contract Vulnerability Detection Via Deep Learning and Data Fusion," *IEEE Transactions on Reliability*, pp. 1–15, 2024.

[6] Z. Wan, X. Xia, D. Lo, J. Chen, X. Luo, and X. Yang, "Smart Contract Security: A Practitioners' Perspective," in *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, 2021, pp. 1410–1422.

[7] L. A. C. Ahakonye, G. C. Amaizu, C. I. Nwakanma, J. M. Lee, and D.-S. Kim, "Classification and Characterization of Encoded Traffic in SCADA Network using Hybrid Deep Learning Scheme," *Journal of Communications and Networks*, vol. 26, no. 1, pp. 65–79, 2024.

[8] Z. Liu, P. Qian, J. Yang, L. Liu, X. Xu, Q. He, and X. Zhang, "Rethinking Smart Contract Fuzzing: Fuzzing With Invocation Ordering and Important Branch Revisiting," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1237–1251, 2023.