Leveraging Expert Rule-Labeling for Smart Contract Vulnerability Detection with BELT

Muhammad Sannan Khaliq [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, loveahakonye, ljmpaul, dskim)@kumoh.ac.kr

***Abstract*—Smart contract vulnerabilities present significant challenges to blockchain security and reliability. This paper introduces a hybrid detection framework that integrates expert rule-based labeling with transformer-based multi-label classification for Solidity smart contracts. Expert rules generate high-quality training labels across key vulnerability types, while BERT embeddings enable robust semantic modeling of complex source code, facilitating accurate detection of vulnerabilities. Experimental results on a large, expert-labeled dataset demonstrate strong performance in terms of micro/macro F1-score, hamming score, and per-category metrics. The approach improves annotation quality and generalization, supporting scalable and accurate automated smart contract auditing.**

***Index Terms*—BiLSTM, BERT, CNN, Expert Rules, Smart Contract, Vulnerability Detection**

I. Introduction

Smart contracts deliver automated logic and value exchange on blockchain platforms, but critical code vulnerabilities have resulted in substantial financial losses and system failures [1], [2]. Common flaws, such as arithmetic errors, unchecked calls, time manipulation, denial-of-service attacks, and reentrancy, require robust detection to secure decentralized applications.

Traditional detection methods leverage either static/dynamic analysis or supervised learning on manually labeled datasets [3], [4]. However, manual annotation is costly, and learning models are heavily dependent on the quality of labels and code complexity [5], [6]. Transformer-based techniques, such as BERT, have recently demonstrated strong performance in smart contract vulnerability detection, outperforming classic static tools across multiple vulnerability categories [7], [8].

We propose a hybrid framework that integrates expert rule-based labeling with transformer-based modeling. Rules provide precise, explainable supervision, while BERT embeddings enable scalable, accurate multi-label classification. This combination delivers reliable detection across diverse contracts, enhancing smart contract auditing.

II. Proposed System

The system integrates expert rule-based labeling with transformer-based deep learning for scalable, multi-label vulnerability detection in Solidity smart contracts. It consists of an expert labeling engine and a classification pipeline. Let the dataset be defined in Equation 1.

$$D = \{(x_i, y_i)\}_{i=1}^{N}, \tag{1}$$

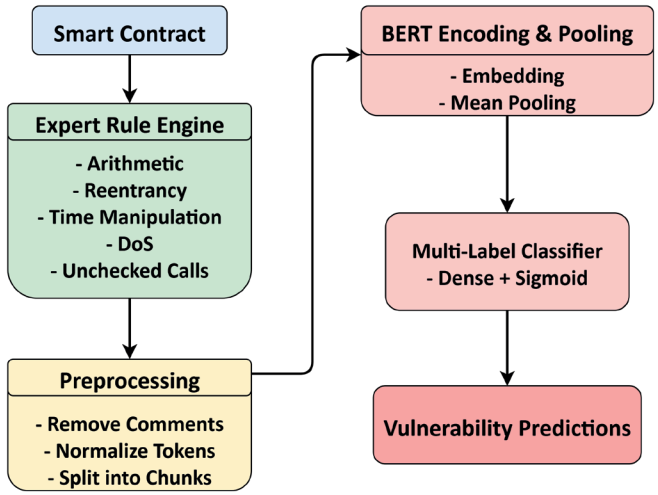

Fig. 1: Proposed System for Smart Contract Vulnerability Detection Using BELT Framework

where x_i denotes a smart contract and $y_i \in \{0,1\}^K$ is a binary label vector indicating the presence or absence of $K = 5$ distinct vulnerabilities from the DASP vulnerability categories. The expert rules systematically analyze each smart contract x_i and assign high-quality annotations y_i, which are subsequently utilized for training and evaluation. Before analysis, contract sources undergo preprocessing: all comments, pragmas, and reserved Solidity keywords are removed, and identifiers are normalized. Each preprocessed contract is segmented into chunks compatible with the input size limit $L_{\max}$ of the BERT model, ensuring even extensive contracts are completely processed. Formally, given a contract x_i, we obtain chunks in Equation 2.

$$\{c_{i,j}\}_{j=1}^{T_i}, \quad \text{where each} \quad |c_{i,j}| \le L_{\max}. \tag{2}$$

Each chunk is embedded independently using a BERT encoder in Equation 3.

$$h_{i,j} = f_{\text{BERT}}(c_{i,j}), \quad h_{i,j} \in \mathbb{R}^d. \tag{3}$$

To aggregate these embeddings into a contract-level representation h_i, we employ average pooling in Equation 4.

$$h_i = \frac{1}{T_i} \sum_{j=1}^{T_i} h_{i,j}. \tag{4}$$

A multi-label classification layer outputs the prediction scores $\hat{y}_i$ in Equation 5.

$$\hat{y}_i = \sigma(W h_i + b), \tag{5}$$

where $\sigma(\cdot)$ denotes the sigmoid activation function. A vulnerability class k is predicted as present if $\hat{y}_i^{(k)} > 0.5$. The training process minimizes the binary cross-entropy loss in Equation 6.

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \left[y_i^{(k)} \log(\hat{y}_i^{(k)}) + (1 - y_i^{(k)}) \log(1 - \hat{y}_i^{(k)}) \right]. \tag{6}$$

For unseen contracts, the multi-label approach enables the detection of overlapping vulnerabilities. System performance is assessed through micro- and macro-F1 scores, Hamming score, and category-specific metrics. Overall, the hybrid architecture integrates domain expertise with transformer-based modeling to deliver scalable and robust vulnerability detection for Solidity smart contracts.

III. Performance Evaluation

Experiments are conducted using the soliaudit-va-dataset-sourcecode, which comprises over 18,000 Solidity contracts labeled by expert rules for five categories of DASP vulnerabilities. The data is split into training, validation, and test sets. The BELT pipeline employs DistilBERT embeddings, mean pooling, and a multi-label classifier. Hyperparameters are tuned using a grid search on the validation F1-score. Performance is assessed using standard multi-label metrics, including the Hamming score and per-category precision, recall, and F1-score, as shown in Table 1.

TABLE I: BELT Model Performance by Category

Category	Precision	Recall	F1-score
Arithmetic	0.97	0.99	0.98
Unchecked_Low_Level_Calls	0.78	0.95	0.85
Time_Manipulation	0.96	0.93	0.95
DoS	0.96	0.84	0.90
Reentrancy	0.93	0.89	0.91

Subset Accuracy is 0.78, reflecting the proportion of contracts where all predicted vulnerability labels exactly match ground truth. Hamming Score (0.89) captures the average label-based agreement using the Jaccard similarity:

$$\text{Hamming Score} = \frac{1}{N} \sum_{i=1}^{N} \frac{|Y_i \cap \hat{Y}_i|}{|Y_i \cup \hat{Y}_i|}, \tag{7}$$

This metric is considered highly informative for multi-label tasks, as it awards partial credit for samples where some, but not all, vulnerabilities are correctly detected. Our model's high Hamming score indicates that the predicted sets closely match the expert annotations, even though subset accuracy is lower due to the strict matching requirements. These results validate the effectiveness of our hybrid expert rule combined with a deep learning approach. High per-category F1 indicates robust overall performance, while a strong Hamming score confirms the reliability of multi-label predictions for practical contract auditing.

IV. Conclusion

We presented a hybrid framework for smart contract vulnerability detection that combines expert rule-based labeling with transformer-based multi-label classification. Leveraging precise expert annotations alongside advanced BERT embeddings enables the robust and scalable detection of multiple vulnerability types within Solidity code. Experimental results on an expert-labeled corpus demonstrate strong performance across micro- and macro-F1, Hamming score, and per-category metrics, confirming the system's effectiveness for practical security auditing. The approach successfully addresses annotation quality and model generalization, advancing the reliability of smart contract analysis. Future work includes expanding expert rules, integrating semi-supervised learning, and applying the framework to additional blockchain platforms and more complex vulnerability types.

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] N. Hejazi and A. H. Lashkari, "A Comprehensive Survey of Smart Contracts Vulnerability Detection Tools: Techniques and Methodologies," *Journal of Network and Computer Applications*, p. 104142, 2025.

[2] H. Chu, P. Zhang, H. Dong, Y. Xiao, S. Ji, and W. Li, "A Survey on Smart Contract Vulnerabilities: Data Sources, Detection and Repair," *Information and Software Technology*, vol. 159, p. 107221, 2023.

[3] W. Deng, H. Wei, T. Huang, C. Cao, Y. Peng, and X. Hu, "Smart Contract Vulnerability Detection Based on Deep Learning and Multimodal Decision Fusion," *Sensors*, vol. 23, no. 16, p. 7246, 2023.

[4] M. S. Khaliq, L. A. C. Ahakonye, J. M. Lee, and D.-S. Kim, "Hybrid DeBERTa-BiLSTM-CNN for Enhanced Smart Contract Vulnerability Detection," in *Proceedings of Symposium of The Korean Institute of Communications and Information Sciences Summer Conference (KICS SUMMER 2025)*, 06 2025.

[5] P. Su and J. Hu, "Smart Contract Vulnerabilities Detection with Bidirectional Encoder Representations from Transformers and Control Flow Graph," *Multimedia Systems*, vol. 30, no. 4, p. 204, 2024.

[6] E. C. Nkoro, L. A. C. Ahakonye, and D. S. Kim, "Explainable DNN for Smart Contract Vulnerability Detection in the Metaverse," *High-Confidence Computing*, 2025.

[7] S. Jeon, G. Lee, H. Kim, and S. S. Woo, "Smartcondetect: Highly Accurate Smart Contract Code Vulnerability Detection Mechanism using BERT1," in *KDD workshop on programming language processing*, vol. 16, 2021, p. 225.

[8] J. Bu, W. Li, Z. Li, Z. Zhang, and X. Li, "Smartbugbert: BERT-Enhanced Vulnerability Detection for Smart Contract Bytecode," *arXiv preprint arXiv:2504.05002*, 2025.