Reading Response 2: Related Work Comparison

CSEC 720 (2225) - Mehul Sen

Research papers, "Neural Nets Can Learn Function Type Signatures From Binaries" (EKLAVYA)[1] by ZL Chua, S Shen, P Saxena, and Z Liang, as well as "Neural Network-based Graph Embedding for Cross-Platform Binary Code Similarity Detection" (Gemini)[2] by Xu et al., present innovative systems that leverage neural networks on binary data to perform operations. However, these papers address different problems, "EKLAVYA" leverages neural networks to recover semantic information such as type recovery from binaries, while "Gemini" employs a neural network to compute an embedding for use in similarity detection between binaries.

The papers differ in terms of how they utilize related work to advance their research objectives. "EKLAVYA" leverages related works to establish relevant research facts before proceeding with the implementation of its neural network. To achieve this, the paper draws from multiple sources, including ECR Shin, D. Song, and R. Moazzezi[3] and Dennis et al.[4], while also relying on the methods and approaches defined by T. Mikolov et al.[5] and K. Simoyan et al.[6] as a foundation for its research. In contrast, "Gemini" uses "Genius", the state-of-the-art approach developed by Feng et al.[7]. Gemini compares its approach with Genius, highlighting the improved performance and accuracy achieved by its method. These papers should differ this way as "EKLAVYA" needs to establish the necessary facts through relevant research before it can proceed with its implementation of the neural network. "Gemini" on the other hand, needs a reference point to assess its performance and effectiveness in executing similarity detection.

Another difference between the related works in these two papers is the extent to which they discuss prior work, existing implementations, and comparisons made in the results. "Gemini" provides a detailed explanation of previous work in the problem domain, the techniques used by the authors to implement Gemini, and why other works in binary code similarity detection cannot directly implement pre-existing approaches. In contrast, "EKLAVYA" lacks such details about existing techniques or any prior work done in this problem domain. For instance, when discussing the evaluation of their system, Eklavya mentions the work of ElWazeer et al.[8] but does not compare their results directly with Eklavya's because their results adopt a different measure of accuracy. On the other hand, the authors of Gemini implement Genius in TensorFlow and directly compute similarity scores while implementing the codebook and embedding generation themselves, enabling precise comparison of the system's performance. Such comparisons and evaluations of related work are essential for establishing the novelty and credibility of the research paper and identifying any limitations while providing context for the research and how it fits in the problem domain.

References

[1]: "Neural Nets Can Learn Function Type Signatures From Binaries." Zheng Leong Chua, Shiqi Shen, Prateek Saxena, and Zhenkai Liang, *USENIX Security Symposium*, Aug. 2017.

[2]: "Neural Network-based Graph Embedding for Cross-Platform Binary Code Similarity Detection." Xiaojun Xu, Chang Liu, Quian Feng, Heng Yin, Le Song, and Dawn Song, *Proceedings of ACM Conference on Computer and Communications Security (CCS'17)*, Oct. 2017.

[3]: "Recognizing Functions in Binaries with Neural Networks." Eui Chul Richard Shin, Dawn Song, and Reza Moazzezi, *USENIX Security Symposium*, Aug. 2015.

[4]: "An In-Depth Analysis of Disassembly on Full-Scale x86/x64 Binaries." Dennis Andriesse, Xi Chen, Victor van der Veen, Asia Slowinska, and Herbert Bos, *USENIX Security Symposium*, Aug. 2016.

[5]: "Linguistic Regularities in Continuous Space Word Representations." Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig, *Proceedings of NAACL-HLT*, Jun. 2013.

[6]: "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps." Karen Simoyan, Andrea Vedaldi, and Andrew Zisserman, *ICLR Workshop*, Apr. 2014.

[7]: "Scalable Graph-based Bug Search for Firmware Images." Qian Feng, Rundong Zhou, Chencheng Xu, Yao Cheng, Brian Testa, and Heng Yin, *Proceedings of ACM Conference on Computer and Communications Security (CCS'16)*, Oct. 2016.

[8]: "Scalable Variable and Data Type Detection in a Binary Rewriter." Khaled ElWazeer, Kapil Anand, Aparna Kotha, Matthew Smithson, and Rajeev Barua, *ACM-SIGPLAN Symposium on Programming Language Design and Implementation*, Jun. 2013.

Appendix

ChatGPT, developed by OpenAI, was utilized to brainstorm and enhance the quality of the paper.

Prompts: "Improve the grammar and flow of this paragraph", "List the benefits of comparisons of related work in a research paper"