Vulnerability Detection via Topological Analysis of Attention Maps

Snopov P.[1], Golubinsky A.N.[1]

Institute for Information Transmission Problems of Russian Academy of Sciences
snopov@iitp.ru

Abstract. Recently, DL-based approaches to the vulnerability detection task became popular. They show promising results, outperforming traditional static code analysis tools.
In this work, we test a novel approach to the vulnerability detection using topological data analysis of the attention matrices of BERT model. We show that ...

Keywords: Vulnerability detection, Persistent homology, Large language models

1 Introduction

The problem of the source code vulnerability detection becomes an important problem in the field of software engineering nowadays. This is a complex task involving the analysis of both syntax and semantics of the source code. The static analysis methods were dominating during most of the time in the vulnerability detection. The static tools tend to rely heavily only on the syntax of the program, thus loosing the potential information that is hidden in the semantics. Therefore such approaches suffer from high rates of false positives.

But recently the different ML-based and DL-based methods started to appear [11,12]. Most DL-based solutions provide the methods based on graph neural networks (GNNs) and large language models (LLMs). Both GNN-based and LLM-based methods are targeted to learn the semantics together with the syntax of the program. This approaches show the promising results, beating the traditional static analysis tools and demonstating low rates of false positives as well as false negatives [6].

Our work demonstates a novel approach to the vulnerability detection problem. As mentioned above, the superiority of DL-based models over static ones are potentially due to the use of the semantic information of the program. This semantic information is captured somehow with neural networks during training or fine-tuning. Moreover, LLM-based models should capture this information even wihout any fine-tune.

On the other hand, as well as with the syntax of the source code, the semantics can also be represented as a tree or a graph. Such representation opens the doors for the use of topological methods.

In this work, we leverage the BERT model, pretrained on the source code, for capturing the semantic information of the programs, and we also use the tools from topological data analysis [3] to retrieve the relations in this semantics. This is an attempt to analyse the interpret the attention matrices via topological instruments.

Our work is inspired by the works of Laida Kushareva et. al. [7], in which the authors apply the topological methods for the artificial text detection, and show that the topology of the semantics captured by a LLM model (such as BERT) provides enough information for the successful classification, beating neural baselines and performing on par with a fully fine-tuned BERT model, but being more robust towards the unseen data.

2 Background

2.1 BERT Model

BERT is a transformer-based language model that has pushed state-of-the-art results in many NLP tasks. The BERT architecture comprises L encoder layers with H attention heads in each layer. The input of each attention head is a matrix X consisting of the d-dimensional representations (row-wise) of m tokens, so that X is of shape $m \times d$. The head outputs an updated representation matrix X^{out}:

$$X^{\text{out}} = W^{\text{attn}}(XW^V), \text{ with } W^{\text{attn}} = \text{softmax}\Big(\frac{(XW^Q)(XW^K)^T}{\sqrt{d}}\Big) \tag{1}$$

where W^Q, W^K, W^V are trained projection matrices of shape $d \times d$ and W^{attn} is of shape $m \times m$ matrix of attention weights. Each element w^{attn}_{ij} can be interpreted as a weight of the j-th input's *relation* to the i-th output: larger weights mean stronger connection between the two tokens.

2.2 Attention Graph

An *attention graph* is a weighted graph representation of an attention matrix W^{attn}, in which the vertices represent the tokens and the edges connect a pair of tokens if the corresponding weight is higher than the predefined threshold value.

This threshold value is set to distinguish weak and strong relations between tokens. But the choice of the threshold value seems to be a very hard problem. Moreover, varying the threshold, the graph structure may change dramatically.

Hopefully, TDA methods can extract the properties of graph structure, without specifying the concrete value of the threshold.

2.3 Topological Data Analysis

Topological data analysis is a young and rapidly evolving field that applies some of the powerful methods from algebraic topology to data science. There are already exist a plethora of good tutorials and surveys for non-mathematicians [8,3] as well as for those who has a mathematical background [4,9,10].

The main instrument in topological data analysis, *persistent homology*, allows one to track the changes in the topological structure for different objects, such as point clouds, scalar functions, images, weighted graphs [1,2].

Specifically, in our work, we are given with a set of tokens V and an attention matrix W. Next, we build a family of attention graphs, which is being indexed by increasing threshold values This family is called a *filtration*, and it is a crucial object in TDA.

When such seqeunce of graphs is given, the persistent homology in dimension 0 and 1 are computed. Dimension 0 illustrates if there are connected components or clusters in our data, and dimension 1 shows if our data possesses cycles or «loops». These calculations provide us with the *persistence diagram*, which can be further used to derive specific topological features, such as the amount of connected components or the amount of cycles (such features are also called the *Betti numbers*).

3 Topological Features of the Attention Graphs

For each code sample, we calculate the persistent homology in dimension 0 and 1 of the symmetrized attention matrices, obtaining the persistence diagram on each attention head of the BERT model. We compute the following features in each dimension from the diagrams:

- The mean lifespan of points on the diagram
- The variance lifespan of points on the diagram
- The max lifespan of points on the diagram
- The overall number of points on the diagram
- The persistence entropy

We symmetrize attention matrices in order to be able to use the persistent homology machinery. When attention matrices are symmetrized, one can think of attention matrices as the distance matrices of some point cloud embedded in Euclidian space. We symmetrize attention matrices in the following manner:

$$\forall i, j : W_{ij}^{\text{sym}} = \max\left(W_{ij}^{\text{attn}}, W_{ji}^{\text{attn}}\right). \tag{2}$$

Alternatively, one can think of attention graphs, in which an edge between the vertices i and j appears when the threshold is greater than both W_{ij}^{attn} and W_{ji}^{attn}.

We consider these features as the numerical characteristic of the semantic evolution processes in the attention heads. These features encode the information about the clusters of mutual influence of the tokens in the sentence and the local structures like cycles. The features with «significant» persistence (i.e. those with large lifespan) correspond to the stable processes, whileas the features with short lifespan are highly influenced to noise and doesn't reflect the stable topological attributes.

4 Experiments

Methodology In order to test whether the topological information encode can be used for vulnerability detection, we train both SVM and LightGBM classifier over topological features derived from the attention matrices from the BERT model as described in Section 3. The training details are outlined in Appendix.

Data We train and evaluate our classifier on *Devign* dataset. This dataset is collected from 2 large C-language open-source projects that are popular among developers and diversified in functionality, i.e., QEMU, and FFmpeg. Due to our computational resources, we were only using those data samples, that, being tokenized, are of length less than 150. In this case, the point cloud that is being constructed during the attention symmetrization, is bounded to be length less than 150.

Baselines We use `microsoft/codebert-base` [5] model from the HuggingFace library [13] as the pre-trained on code dataset BERT-based baseline. We also fully fine-tune `microsoft/codebert-base`.

5 Results and Discussion

Table 1 outlines the results of the vulnerability detection experiments on the *Devign* dataset. The results reveal that the proposed topology-based classifiers ourperform chosen LLM without fine-tuning but perform worse than the fine-tuned version.

Table 1. The results of the vulnerability detection experiments.

Model	F1 score	Accuracy
Logistic Regression	0.22	0.54
LightGBM	**0.55**	0.63
SVM	0.54	**0.65**
CodeBERTa (pre-trained)	0.28	0.45
CodeBERTa (fine-tuned)	**0.71**	0.72

This observation shows that the information about the code snippet being vulnerable is encoded in the topological attributes of the attention matrices. The semantic evolution in the attention heads reflects the code properties that happen to be important in the vulnerability detection task. And persistent homology appears to be a suitable method for extracting this information.

Notice that only the semantic information from attention heads was used. The use of additional topological features obtained from the structural information of the source code, such as the topology of the graph representations of the source code, might probably increase the overall quality of the proposed models.

6 Consclusion

This paper introduces a novel approach for the vulnerability detection task based on TDA. We propose the set of interpretable topological features, obtained from the persistence diagrams, that are derived from the attention matrices of any transformer-based LM. The experiments show that the ML classifiers, trained on these features, perform on par with pre-trained on code LLM.

We are publicly releasing our code, hoping to stimulate the research on the TDA-based methods to vulnerability detection and other NLP tasks. A potential direction for future work is to combine topological features that encode semantics with the topological features that encode structural information. It is also interesting to investigate the topology of different symmetrisations of the attention matrices and how they encode the semantics. Another interesting direction is to incorporate multiparameter persistent homology to the study of the semantic evolution in the attention heads.

Acknowledgements The work was carried out as part of the state assignment N 1021061609772-0-1.2.1 (FFNU-2024-0020).

References

1. Adams, H., Chepushtanova, S., Emerson, T., Hanson, E., Kirby, M., Motta, F., Neville, R., Peterson, C., Shipman, P., Ziegelmeier, L.: Persistence images: A stable vector representation of persistent homology (2016), https://arxiv.org/abs/1507.06217
2. Aktas, M.E., Akbas, E., Fatmaoui, A.E.: Persistence homology of networks: methods and applications. Applied Network Science **4**(1) (Aug 2019). https://doi.org/10.1007/s41109-019-0179-3, http://dx.doi.org/10.1007/s41109-019-0179-3
3. Chazal, F., Michel, B.: An introduction to topological data analysis: Fundamental and practical aspects for data scientists. Frontiers in Artificial Intelligence **4** (2021). https://doi.org/10.3389/frai.2021.667963, https://www.frontiersin.org/articles/10.3389/frai.2021.667963
4. Edelsbrunner, H., Harer, J.: Persistent homology—a survey (2008). https://doi.org/10.1090/conm/453/08802, http://dx.doi.org/10.1090/conm/453/08802
5. Feng, Z., Guo, D., Tang, D., Duan, N., Feng, X., Gong, M., Shou, L., Qin, B., Liu, T., Jiang, D., Zhou, M.: Codebert: A pre-trained model for programming and natural languages (2020)
6. Katsadouros, E., Patrikakis, C.: A Survey on Vulnerability Prediction using GNNs. In: Proceedings of the 26th Pan-Hellenic Conference on Informatics. pp. 38–43. ACM, Athens Greece (nov 2022). https://doi.org/10.1145/3575879.3575964, https://dl.acm.org/doi/10.1145/3575879.3575964
7. Kushnareva, L., Cherniavskii, D., Mikhailov, V., Artemova, E., Barannikov, S., Bernstein, A., Piontkovskaya, I., Piontkovski, D., Burnaev, E.: Artificial Text Detection via Examining the Topology of Attention Maps. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 635–649 (2021). https://doi.org/10.18653/v1/2021.emnlp-main.50, http://arxiv.org/abs/2109.04825, arXiv:2109.04825 [cs, math]

8. Murugan, J., Robertson, D.: An introduction to topological data analysis for physicists: From lgm to frbs (2019), https://arxiv.org/abs/1904.11044
9. Oudot, S.: Persistence Theory: From Quiver Representations to Data Analysis. American Mathematical Society (Dec 2015). https://doi.org/10.1090/surv/209, http://dx.doi.org/10.1090/surv/209
10. Schenck, H.: Algebraic Foundations for Applied Topology and Data Analysis. Springer International Publishing (2022). https://doi.org/10.1007/978-3-031-06664-1, http://dx.doi.org/10.1007/978-3-031-06664-1
11. Sharma, T., Kechagia, M., Georgiou, S., Tiwari, R., Vats, I., Moazen, H., Sarro, F.: A survey on machine learning techniques applied to source code. Journal of Systems and Software **209**, 111934 (mar 2024). https://doi.org/10.1016/j.jss.2023.111934, https://www.sciencedirect.com/science/article/pii/S0164121223003291
12. Steenhoek, B., Rahman, M.M., Jiles, R., Le, W.: An Empirical Study of Deep Learning Models for Vulnerability Detection (feb 2023). https://doi.org/10.48550/arXiv.2212.08109, http://arxiv.org/abs/2212.08109, arXiv:2212.08109 [cs]
13. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T.L., Gugger, S., Drame, M., Lhoest, Q., Rush, A.M.: Huggingface's transformers: State-of-the-art natural language processing (2020), https://arxiv.org/abs/1910.03771