BSP Project Description: Malicious URL Detection with Large Language Models

Ivaylo Krumov
University of Luxembourg
Email: ivaylo.krumov.001@student.uni.lu

This report has been produced under the supervision of:
Salima Lamsiyah
University of Luxembourg
Email: salima.lamsiyah@uni.lu

Abstract

This document outlines the end of Phase I description of the project carried out by Ivaylo Krumov, under the guidance of Salima Lamsiyah, during his Bachelor Semester Project 6. It is about the ability and application of large language models in helping prevent URL-based cyber attacks by identifying malicious web address links. The project will detail the scientific aspects, consisting of the acquisition of knowledge relevant to the scientific question at hand and to the development of the technical deliverable, and the technical aspects, consisting of the application of the gathered knowledge about the project's main topic.

1. Plagiarism statement

I declare that I am aware of the following facts:

- As a student at the University of Luxembourg I must respect the rules of intellectual honesty, in particular not to resort to plagiarism, fraud or any other method that is illegal or contrary to scientific integrity.
- My report will be checked for plagiarism and if the plagiarism check is positive, an internal procedure will be started by my tutor. I am advised to request a pre-check by my tutor to avoid any issue.
- As declared in the assessment procedure of the University of Luxembourg, plagiarism is committed whenever the source of information used in an assignment, research report, paper or otherwise published/circulated piece of work is not properly acknowledged. In other words, plagiarism is the passing off as one's own the words, ideas or work of another person, without attribution to the author. The omission of such proper acknowledgement amounts to claiming authorship for the work of another person. Plagiarism is committed regardless of the language of the original work used. Plagiarism can be deliberate or accidental. Instances of plagiarism include, but are not limited to:
  1) Not putting quotation marks around a quote from another person's work
  2) Pretending to paraphrase while in fact quoting
  3) Citing incorrectly or incompletely
  4) Failing to cite the source of a quoted or paraphrased work
  5) Copying/reproducing sections of another person's work without acknowledging the source
  6) Paraphrasing another person's work without acknowledging the source
  7) Having another person write/author a work for oneself and submitting/publishing it (with permission, with or without compensation) in one's own name ('ghost-writing')
  8) Using another person's unpublished work without attribution and permission ('stealing')
  9) Presenting a piece of work as one's own that contains a high proportion of quoted/copied or paraphrased text (images, graphs, etc.), even if adequately referenced

  Auto- or self-plagiarism, that is the reproduction of (portions of a) text previously written by the author without citing that text, i.e. passing previously authored text as new, may be regarded as fraud if deemed sufficiently severe.

2. Main required competencies

The following is a description of the specific scientific and technical competencies, that are essential for the execution and success of this project:

2.1. Scientific main required competencies

The main scientific competencies required for this project include adequate knowledge in the fields of cybersecurity and cyber threat intelligence, particularly in the practical use of large language models for malicious URL detection. This mainly involves an understanding of natural language

processing techniques, including text classification and pattern recognition, knowledge about the concept of large language models and their inner workings and familiarity with existing work that deals with the detection of malicious web addresses.

2.2. Technical main required competencies

The main technical competencies needed for a successful execution of the project include fluency in Python, especially in its application to cybersecurity tasks such as URL classification, which includes a practical understanding of natural language processing and machine learning techniques. Knowledge in data processing and analysis is crucial for parsing and examining URL datasets. Additionally, a grasp of basic data visualization skills is essential for identifying any existing patterns and anomalies in data related to malicious websites. Finally, a basic understanding of model performance is required to analyze and interpret the effectiveness of models in detecting malicious URLs.

3. Scientific Deliverable Description

The main scientific aspect of this project includes gaining a comprehensive understanding in the field of machine learning, particularly focusing on the fine-tuning of large language models (LLMs). More specifically, the scientific deliverable will aim to explore the utility that LLMs provide in the area of cyber threat intellingence, particularly for the task of classifying web address URLs as benign or malicious. This involves an in-depth study of machine learning principles, including model architecture and parameter optimization, familiarity with the techniques for adapting LLMs to URL classification through fine-tuning, and the ability to evaluate the performance of these models effectively. On a deeper level, it is also vital to take a look into the theoretical basis of natural language processing and machine learning algorithms. This understanding will aim to provide a solid foundation for the subsequent technical deliverable and ultimately giving an answer to the scientific question: "How can we use large language models to detect malicious URLs?". To achieve this, several scientific papers and articles will be utilized, which will serve as a knowledge base for the formulation of the final answer. The following is a list of the referenced resources that are expected to contribute towards this goal (to be potentially expanded at a later point in time):

- "Lexical features based malicious URL detection using machine learning techniques." by [Raja, A. Saleem, R. Vinodini, and A. Kavitha (2021)]
- "Malicious URL detection using machine learning: a survey." by [Doyen, S. (2019)]
- "URLNet: Learning a URL representation with deep learning for malicious URL detection." by [Le, H., Pham, Q., Sahoo, D., & Hoi, S. C. (2018)]
- "What's in a url: Fast feature extraction and malicious url detection." by [Verma, R., & Das, A. (2017, March)]
- "Detecting malicious URLs using machine learning techniques: review and research directions." by [Aljabri, M., Altamimi, H. S., Albelali, S. A., Al-Harbi, M., Alhuraib, H. T., ... (2022)]
- "Malicious URL detection using convolutional neural network." by [Abdi, F. D., & Wenjuan, L. (2017)]

The primary focus of the scientific deliverable is to introduce key concepts in natural language processing, machine learning and LLMs and their utility in the context of cyber threat intelligence, which is crucial to addressing the aforementioned scientific question.

First of all, the deliverable will give a brief introduction to the current state of cybersecurity and cyber threat intelligence, given that these fields are directly related to the topic of the scientific question. This section will be followed by a closer look at the practical cases of utilizing web addresses for malicious purposes, while also highlighting the importance of malicious URL detection in preventing such cyber attacks.

Next, the concept of using LLMs for the detection of malicious URLs will be introduced. The section will cover the theoretical foundations of LLMs and how they have already been applied in existing work in natural language processing and cybersecurity. It will specifically focus on the process of fine-tuning LLMs to recognize the subtle patterns and linguistic cues that distinguish malicious URLs from benign ones.

Furthermore, it will be essential to understand how certain algorithms, commonly used in natural language processing and machine learning tasks, work - both in theory and in practice. This exploration will give further insight into the possibilities for a LLM to be fine-tuned particularly for the use of URL classification.

Once all these topics have been covered, the scientific deliverable will conclude with a formulation of an answer to the proposed scientific question. The answer will be derived from the gathered knowledge in the previous sections of the deliverable, prioritizing information about type of training and algorithms used for LLMs and how the accuracy in the task to classify URLs as either benign or malicious compares between different models. This technical aspect to the question's answer will further serve as a basis to justify the method that will be used in the development of the technical deliverable.

4. Technical Deliverable Description

The technical deliverable of this project will combine the practical aspects and application of the principles explored in the scientific deliverable. More specifically, it will consist of a Python implementation and evaluation of at least two LLMs tailored for URL classification, each using different training methods and/or parameters. The coding environment of choice for these tasks will be a Jupyter notebook, as it offers plenty of flexibility and code readability suitable for machine learning problems.

The developed models will be trained and evaluated using the same publicly available dataset contain-

ing a large mix of URLs labeled as either benign or malicious [https://github.com/Priyanshu9898/End-to-End-Malicious-URL-Detection]. The selected dataset will be pre-processed accordingly for convenience and simplification with relevant libraries such as Numpy, if needed. The two main techniques planned for the training of the models will be standard learning with cross-entropy loss function and reinforcement learning. For the latter, the Transformer Reinforcement Learning (TRL) library will be used, as it is able to provide a wide range of sophisticated algorithmic and optimization tools specialized for training LLMs with reinforcement learning. The evaluation step will analyze and compare the performance of the different models by metrics such as accuracy, F1 score and precision, as well as by plotting suitable visualizations, making use of the relevant utilities offered by the Scikit-learn and Matplotlib libraries.

The entire implementation process of the models will be explained in detail in the final technical deliverable, justifying the methods and techniques used, mainly based on acquired knowledge from the scientific deliverable. The chosen dataset and aspects of the selected libraries, such as some of TRL's utilities, will also be detailed, as they will play a major role in the overall development process. Finally, the interpretation of the produced results and their comparison will be documented alongside a brief concluding discussion of what they imply for the practical possibility of applying LLMs in identifying malicious URLs.

References

[Raja, A. Saleem, R. Vinodini, and A. Kavitha (2021)] "Lexical features based malicious URL detection using machine learning techniques." Materials Today: Proceedings 47 (2021): 163-166.

[Doyen, S. (2019)] Malicious URL detection using machine learning: a survey. ArXiv, 1701, v3.

[Le, H., Pham, Q., Sahoo, D., & Hoi, S. C. (2018)] URLNet: Learning a URL representation with deep learning for malicious URL detection. arXiv 2018. arXiv preprint arXiv:1802.03162.

[Verma, R., & Das, A. (2017, March)] What's in a url: Fast feature extraction and malicious url detection. In Proceedings of the 3rd ACM on International Workshop on Security and Privacy Analytics (pp. 55-63).

[Aljabri, M., Altamimi, H. S., Albelali, S. A., Al-Harbi, M., Alhuraib, H. T., ... (2022)] Detecting malicious URLs using machine learning techniques: review and research directions. IEEE Access, 10, 121395-121417.

[Abdi, F. D., & Wenjuan, L. (2017)] Malicious URL detection using convolutional neural network. Journal International Journal of Computer Science, Engineering and Information Technology, 7(6), 1-8.

[https://github.com/Priyanshu9898/End-to-End-Malicious-URL-Detection]