Fine-tuning BERT for HTTP Payload Classification in Network Traffic

Mayur Sinha
sinhamayur@yahoo.com

Sangram Kesari Ray
shankar.ray030@gmail.com

Khirawadhi
Khirawdhi@gmail.com

December 20, 2023

Abstract

Fine-tuning pre-trained language models like Bidirectional Encoder Representations from Transformers (BERT)[1] has exhibited remarkable potential in various natural language processing[2] tasks. In this study, we propose and investigate the fine-tuning of BERT specifically for the classification of HTTP payload representations within network traffic. Given BERT's adeptness at capturing semantic relationships among tokens, we aim to harness its capabilities for discerning normal and anomalous patterns within HTTP payloads. Leveraging transfer learning by fine-tuning BERT, our methodology involves training the model on a task-specific dataset to adapt its pre-trained knowledge to the intricacies of HTTP payload classification. We explore the process of fine-tuning BERT to learn nuanced representations of HTTP payloads and effectively distinguish between normal and anomalous[3] traffic patterns. Our findings reveal the potential efficacy of fine-tuned BERT models in bolstering the accuracy and efficiency of anomaly detection mechanisms within network communications.

1 Introduction

The surge in sophisticated cyber threats[4] necessitates advanced techniques for anomaly detection within network traffic. Deep learning models, particularly Bidirectional Encoder Representations from Transformers (BERT), have showcased remarkable success in capturing intricate semantic relationships within natural language text. Exploiting BERT's ability to comprehend nuanced linguistic contexts, we propose a pioneering methodology for discerning semantic relationships among token representations extracted from HTTP payloads.
Traditional anomaly[5] detection techniques often rely on pattern matching and statistical analysis, often falling short in capturing complex, context-dependent anomalies prevalent in network traffic. In contrast, our approach harnesses

the power of BERT, allowing us to encapsulate the rich semantic information inherent in HTTP payload tokens. By leveraging BERT's pre-trained contextual embeddings, we endeavor to classify token representations as either normal or anomalous, thereby enhancing the accuracy and efficacy of anomaly detection in network communications.
This paper outlines our methodology, which involves leveraging BERT's language understanding capabilities to decode intricate relationships within HTTP payload token representations. We hypothesize that this novel approach will significantly contribute to the robustness and precision of anomaly detection systems, providing a more sophisticated means of identifying and mitigating network-based threats.

2 Methodology

2.1 Data Collection and Preprocessing

The HTTP dataset CSIC 2010[6] has been used for training and evaluation. Query parameters of the HTTP requests has been extracted from the dataset and preprocessing steps applied, namely urldecoding, lower casing, replacing '=' and '@' characters with white-space, since we're treating the problem at hand as text-classification. Finally, we convert our dataset to a huggingface dataset, because later it'll make tokenization and training easier.

2.2 Tokenization

The choice of Hugging Face's bert-base-uncased tokenizer is particularly significant in our study. It adeptly handles the diverse characteristics of network traffic data, which often lacks consistent capitalization but is rich in contextual and semantic intricacies. This tokenizer not only facilitates the conversion of raw data into a BERT-friendly format but also ensures the nuances of the network traffic are accurately captured and interpreted. The effectiveness of this process is pivotal for the subsequent stages of model training and classification, highlighting the synergistic integration of advanced NLP tools with cybersecurity data.

2.3 Fine-tuning BERT

We're using bert-base-uncased pre-trained model using huggingface AutoModelForSequenceClassification with 2 labels to add a classification head to the BERT model. We're only going to tune the parameters of the classification head during training on our dataset.

2.4 Training Procedure

We're using hugginface Trainer API, with batch size of 8, learning rate of 5e-5, 5 epochs, linear learning rate scheduler. Fine-tuning is done with few epochs

to handle overfitting, in addition to dropout regularization at the classification head.

2.5 Evaluation Metrics and Definitions

Accuracy is used to assess the overall performance of the fine-tuned BERT model.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Accuracy: Accuracy measures the overall correctness of the model's predictions.

True Positive (TP): The number of instances correctly classified as positive by the model.

False Positive (FP): The number of instances incorrectly classified as positive by the model when they are actually negative.

True Negative (TN): The number of instances correctly classified as negative by the model.

False Negative (FN): The number of instances incorrectly classified as negative by the model when they are actually positive

2.6 Experimental Setup

We've split the CSIC data as 70-30 for training and validation respectively.

2.7 Software and Tools

We've used datasets, transformers, and trainer librarier from huggingface. We're using notebook from Google Colab Pro running on 1 V100 GPU.

3 Results

3.1 Performance Metrics

In 5 epochs, our model achieves 86% accuracy.

Epoch	Training Loss	Validation Loss	Accuracy
1	No log	0.619758	0.635000
2	No log	0.483919	0.753000
3	No log	0.432686	0.797000
4	0.477600	0.485009	0.863000
5	0.477600	0.558625	0.864000

Figure 1: Accuracy after 5 epochs

3.2 Discussion of Findings

The 86% accuracy achieved in our study, with only five epochs of training on a smaller dataset, is a testament to the efficiency and potential of BERT's fine-tuning in the cybersecurity domain. This high level of accuracy, attained with limited data and training, highlights BERT's ability to adapt to the intricacies of network traffic, even in complex cybersecurity environments. This result not only validates the approach but also sets a benchmark for future research in applying deep learning models like BERT in similar contexts, offering a promising avenue for enhancing network security protocols against evolving cyber threats.

3.3 Significance and Implications

The application of BERT in HTTP payload classification heralds a significant advancement in cybersecurity. By fine-tuning BERT, we demonstrate a novel, more nuanced approach to anomaly detection, moving beyond traditional, often rigid, rule-based systems. This method shows promise in increasing the precision of anomaly detection, significantly reducing false positives and false negatives, which are common pitfalls in existing systems. Furthermore, the adaptability of BERT to context-specific nuances in network traffic presents an opportunity for more dynamic, responsive cybersecurity mechanisms. This research could lay the groundwork for more intelligent, adaptable cybersecurity systems, potentially transforming how network security is approached in various industries. The implications of this study extend to enhancing the security protocols of critical infrastructures, financial institutions, and other sectors vulnerable to sophisticated cyber threats.

4 Data Availability

The code can be downloaded from here: https://github.com/b1nch3f/fine-tuning-bert-for-http-anomaly

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2018.

[2] D. Jurafsky and J. H. Martin. Speech and language processing. 2020.

[3] Owasp TOP10. Web application security risk. https://owasp.org/www-project-top-ten/. 2021.

[4] WASC. Web application security consortium. http://www.webappsec.org/. 2010.

[5] M. E. H. Holm. Estimates on the effectiveness of web application firewalls against targeted attacks pp. 250–265. 2013.

[6] C. T. Giménez, A. P. Villegas, and G. Á. Marañón. Http dataset csic. 2010.