Technical Report, Version 1

Felix Childress
fdchild22@earlham.edu
Earlham College
Richmond, Indiana, USA

Abstract

Intrusion detection systems (IDS) play a critical role in identifying malicious activity within network traffic, yet traditional methods often rely on models that largely struggle to generalize novel threats. This project explores the application of natural language processing (NLP) techniques, particularly that of transformer-based models such as BERT (Bidirectional Encoder Representations from Transformers), to the domain of network packet analysis for intrusion detection. I propose a tokenization strategy that treats flow-level statistical features as structured sequences, enabling network behavior patterns to be analyzed similarly to natural language. Utilizing the CIC-IDS 2017 dataset, the performance of a fine-tuned BERT model is compared to that of a random forest baseline as a means of assessing the viability of NLP-driven approaches for cybersecurity applications as opposed to traditional methods. Core contributions include a novel tokenization pipeline for converting network flow features into BERT-compatible input, a systematic comparison of traditional and NLP-based detection techniques, and an evaluation of how feature representation and tokenization strategy influences detection performance.

CCS Concepts

• **Security and privacy** → **Intrusion detection systems**; • **Networks** → *Packet classification*; • **Computing methodologies** → **Natural language processing**; *Classification and regression trees*; Supervised learning; • **Computer systems organization** → *Neural networks.*

Keywords

malware, networking, traffic, nlp, ids

ACM Reference Format:
Felix Childress. 2025. Technical Report, Version 1. In . ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Recent years have witnessed an exponential rise in both the frequency and complexity of cyberattacks. As networked systems become more integrated into critical infrastructure, an increasing number of fields and disciplines have come to rely heavily on a continuous and secure exchange of data for operation. These systems depend on the smooth transmission of millions of network packets every second, each one carrying instructions, requests, or sensitive information. Malware targeting these systems attempt to disguise itself as legitimate traffic, making accurate detection a significant challenge.

Traditional intrusion detection systems face significant challenges as the thread landscape continues to evolve. Signature-based detection methods rely on predefined patterns of known attacks, which hinders their ability to identify new threats effectively. Anomaly-based detection methods, while better at handling these unknown attacks, suffer from high false positive rates and have a harder time adapting to changing network behaviors. Further, conventional machine learning methods applied to network security, such as support vector machines and decision trees, tend to treat network flows as independent feature vectors. This has the potential to miss the temporal and sequential patterns that characterize many sophisticated attacks, and is particularly problematic as modern attacks increasingly involve coordinated and multi-step processes.

To address these limitations, the application of various NLP techniques have begun to be explored for cybersecurity applications. As NLP models learn patterns in text by treating words as tokens in meaningful sequences, network traffic analysis could potentially benefit from treating network features as sequential elements with contextual relationships. However, little work exists focusing on adapting transformer-based architectures for network flow analysis. As such, this paper aims to address this gap by analyzing if BERT NLP models can effectively detect network intrusions when applied to tokenized representations of network flow statistics. Using the CIC-IDS 2017 dataset, a widely-used benchmark for intrusion detection research, the current focus is on SSH brute-force attacks; further will be done introducing other attacks, such as FTP and DDoS.

2 Hypothesis

As it stands, my current hypothesis is that, when given appropriately tokenized flow-level network data, a BERT-based NLP approach will achieve a higher overall detection performance (in particular with recall and F1-scores) than a traditional RF classifier trained on the same dataset.

3 Literature Review

3.1 Network Security

Network packets are small, individual units of data broken down from larger messages that are then transmitted across a network. Each packet primarily consists of two components: the header and the payload. The header consists primarily of control information, such as sequencing information, protocol types, and IP addresses

Conference'17, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

for the source and destination; the payload, on the other hand, is the actual, intended message being transmitted. [7] In essence, the purpose of the header is to provide context for the payload and, in a way, acts as the "envelope" for the payload to send it on its way. Network traffic, then, is the flow of packets over a network at any given moment.

3.2 Malware Detection and Analysis

Traditional intrusion detection techniques for identifying malware in a system typically fall into one of two categories: signature-based and anomaly-based detection.

Signature-based detection relies primarily on signatures, which are recognizable patterns or characteristics that are associated with malicious activity in some way. These are extracted from a packet's payload, and from there they're compared to what is known as a signature library—essentially just a database for known or common signatures. An alert is sent forward if a match is found, and the packet is then typically redirected to a separate application that filters or disposes of the packet. [1] This proves effective in the case of malware with payloads that are both consistent and unencrypted; however, if a threat is new and lacking a defined signature, or the payload may be encrypted or otherwise obfuscated, then it is possible that it will slip under the radar. In addition, the packet header is rarely considered when it comes to signature-based detection, and thus can ignore the "red flags" manifesting in metadata.

There is also anomaly-based detection, a machine learning (ML) approach to intrusion detection that flags deviations from expected, normal activity occurring within a system. What is considered "normal activity" for a system can be identified in multiple different ways, but typically always involves analyzing the behavior of a user profile in some way over time and creating a rule-based model with this data that can be later used as a baseline comparison. [2] As opposed to signature-based detection, anomaly-based systems are much better at identifying novel threats, and as many of them are based in ML, accuracy when identifying what is or is not a threat can be improved over time. However, there are still a host of issues. Most of these arise as it is difficult to find attack-free data to train with. If this data includes attacks, any behaviors associated with them that affect the system are often mistakenly trained to be seen as normal, meaning that similar behavior in other attacks might get overlooked. However, if the data is completely attack free, this can lead to the model having an increased sensitivity to any slight change, resulting in a higher rate of false positives for malicious behavior. [8]

3.3 Natural Language Processing

Natural language processing is a subfield of computer science that aims to train computers to process, generate, and manipulate human language. [10] This includes utilizing extensive linguistic knowledge to analyze the overall structure of a language from the ground up, starting at the word level and gradually moving up to the sentence at large and the overall context of a piece of text. [3] From there, this data is typically fed to different ML algorithms that can then create a conceptual model of how it believes a language operates, which can then be used for a variety of tasks, such as generating predictions or classifying content according to their linguistic features. Some common models include Google's BERT, which had been pre-trained using a large corpora of English text for ease of use [5], and their earlier Word2vec, which aims to create vector representations of individual words.

This unique approach to sequencing and processing data offers a unique advantage when it comes to intrusion detection. In the context of network traffic, packet headers can be "tokenized," or split into individual, manipulatable units, and then fed into ML models. [6] These models can then analyze and identify further patterns in how the different components interact both within individual packets and across entire sessions. In addition, packet headers are inherently sequential as a means of ensuring data is being delivered in the correct order. [4] NLP models tend to be well-equipped for this type of data as they are designed to understand and model the relationships between tokenized elements. Just as words in a sentence follow grammatical and syntactical rules that determine their overall order and relationship, the fields of a packet header follow similar structural conventions. Once treated as language-like input, they can be contextually analyzed, which allows for the model to detect subtle deviations from typical traffic patterns that can indicate malicious or otherwise abnormal activity. [11] Unlike traditional signature-based detection, which rely on predefined rules and analyze data in isolation, NLP models, especially those based in ML, can infer new patterns from the given data itself, which makes it both adaptable and naturally well-equipped for detecting previously unseen threats. [9] In regards to anomaly-based detection, which often struggle adapting to changing traffic patterns and distinguishing between malicious and benign activity, NLP models allow for characterization of normal behavior that takes into consideration latent patterns, contextual dependencies, and abstract relationships between elements that can be overlooked otherwise. This in turn allows for a more nuanced understanding of traffic behavior that avoids tripping up the system in the same way. [9]

4 Methods

4.1 Dataset

The dataset being utilized is the CIC-IDS 2017 dataset created by the Canadian Institute for Cybersecurity from the University of New Brunswick. It contains labeled network traffic captures representing both benign and malicious activity across multiple simulated work days and multiple attack types (such as brute forcing and port scanning). Each record corresponds to a bidirectional network flow and features over seventy statistical features representing traffic characteristics (such as packet count and flow duration). Two subsets exist: one PCAP, which simulates realistic, raw traffic data, and a CSV pre-sorted for feasbility. The dataset is available for free use with proper credit. While I aim to integrate all five days worth of data, for the purposes of sorting through kinks and getting preliminary results, data will be added gradually, and all done so far has been trained using the Monday (which is fully benign) and Tuesday data (which has a mix of SSH and FTP attacks; only the former is considered for now) alone.

4.2 Preprocessing

Data preprocessing involved several key steps:

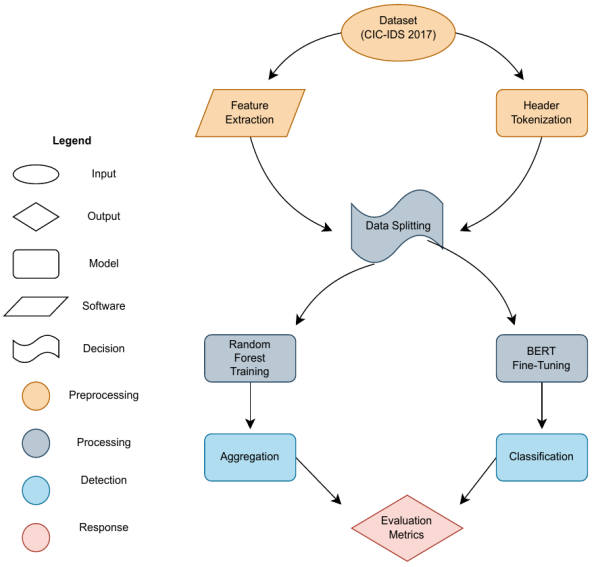

Figure 1: Data Architecture Diagram

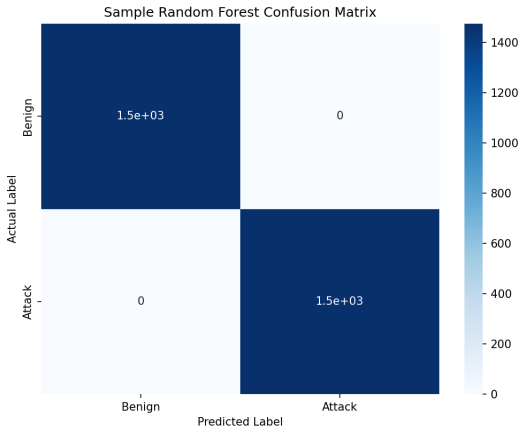

Figure 2: Preliminary RF confusion matrix

(1) Importing the dataset using `pandas` and examined for inconsistent and missing values
(2) Removing features irrelevant to flow-level behavior, such as timestamps and IP addresses
(3) Balancing the data by oversampling attack samples and undersampling benign; this was done due to the high imbalance between benign and attack classes in the dataset
(4) Splitting the processed data into training and testing sets by class label

4.3 Model Design and Training

The random forest classifier was implemented using `scikit-learn` and primarily trained on the normalized feature set, excluding the label. In addition, feature importance scores were extracted post-training to analyze the relative contribution of each flow statistic to model decisions.

For my transformer-based model, the Hugging Face Transformers library is being utilized with a PyTorch backend. I developed a pipeline for tokenization to convert string tokens into feature values. However, I've been having issues in regards to installation of the BERT model itself, and as such while the tokenization script is available, current results for BERT are not.

4.4 Results

Model performance was evaluated using standard classification metrics, mainly accuracy, precision, recall, and F1-score (which acts as the mean). In the future, both the RF and BERT models will be evaluated on the same test set to ensure comparability. In addition, a confusion matrix was plotted for the RF model to visualize class-level performance, which will also be done on BERT.

So far, the RF model shows a `1.00` score for precision, recall, and F1-score. This suggests that an SSH brute-force is fairly simple for an RF classifier to detect, but also that, should this score stays with further testing, my BERT model will have to match completely to show improvement.

5 Future Work

Immediately my next steps are to fix the issues with my BERT installation and configuration in order to fully create a preliminary comparison; from there, creating and revising a graphical abstract. Going forward, however, I also aim to increase the amount of data being used from just Monday and Tuesday to all data from all five days; I also aim to clean up the scripts so that, in addition to having separate scripts to analyze each day as I have been creating, I can have one script/program/notebook to analyze all days at once.

References

[1] Ahmad Azab, Mahmoud Khasawneh, Saed Alrabaee, Kim-Kwang Raymond Choo, and Maysa Sarsour. 2024. Network traffic classification: Techniques, datasets, and challenges. *Digital Communications and Networks* 10, 3 (2024), 676–692.
[2] Monowar H Bhuyan, Dhruba Kumar Bhattacharyya, and Jugal K Kalita. 2013. Network anomaly detection: methods, systems and tools. *Ieee communications surveys & tutorials* 16, 1 (2013), 303–336.
[3] KR1442 Chowdhary and KR Chowdhary. 2020. Natural language processing. *Fundamentals of artificial intelligence* (2020), 603–649.
[4] Glen Gibb, George Varghese, Mark Horowitz, and Nick McKeown. 2013. Design principles for packet parsers. In *Architectures for Networking and Communications Systems*. IEEE, 13–24.
[5] Md Saiful Islam and Long Zhang. 2024. A Review on BERT: Language Understanding for Different Types of NLP Task. *Preprints. org* (2024).
[6] Sabrina J Mielke, Zaid Alyafeai, Elizabeth Salesky, Colin Raffel, Manan Dey, Matthias Gallé, Arun Raja, Chenglei Si, Wilson Y Lee, Benoît Sagot, et al. 2021. Between words and characters: A brief history of open-vocabulary modeling and tokenization in NLP. *arXiv preprint arXiv:2112.10508* (2021).
[7] Leslie F Sikos. 2020. Packet analysis for network forensics: A comprehensive survey. *Forensic Science International: Digital Investigation* 32 (2020), 200892.
[8] Robin Sommer and Vern Paxson. 2010. Outside the closed world: On using machine learning for network intrusion detection. In *2010 IEEE symposium on security and privacy*. IEEE, 305–316.
[9] Zarrin Tasnim Sworna, Zahra Mousavi, and Muhammad Ali Babar. 2023. NLP methods in host-based intrusion detection Systems: A systematic review and future directions. *Journal of Network and Computer Applications* 220 (2023), 103761.
[10] David Okore Ukwen and Murat Karabatak. 2021. Review of NLP-based systems in digital forensics and cybersecurity. In *2021 9th International symposium on digital forensics and security (ISDFS)*. IEEE, 1–9.

[11] Yong Yang and Xing Peng. 2025. BERT-based network for intrusion detection system. *EURASIP Journal on Information Security* 2025, 1 (2025), 11.