BERT-Based Sentiment Forensics Analysis for Intrusion Detection

Shahrzad Sayyafzadeh
Dept. of Comp. & Info.Sciences
Florida A&M University
Tallahassee, FL 32307
shahrzad1.sayyafzade@famu.edu

Hongmei Chi
Dept. of Comp. & Info Sciences
Florida A&M University
Tallahassee, FL 32307
hongmei.chi@famu.edu

Weifeng Xu
School of Criminal Justice
The University of Baltimore,
Baltimore, MD
wxu@ubalt.edu

Kaushik Roy
Dept. of Computer Science
North Carolina A&T State University
Greensboro, NC
kroy@ncat.edu

2023 International Conference on Machine Learning and Applications (ICMLA) | 979-8-3503-4534-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICMLA58977.2023.00235

***Abstract*—The need for effective intrusion detection and user behavior analytics in cybersecurity has reached unprecedented levels. By leveraging the power of BERT, a pre-trained language model known for its contextual understanding, the goal is to uncover latent patterns and insights within textual data to identify potential threats and anomalous user behavior. This study aims to further advance network analysis capabilities by integrating BERT (Bidirectional Encoder Representations from Transformers) for fine-tuning an Long Short-Term Memory (LSTM) model with attention mechanisms. By incorporating attention mechanisms, these models intelligently prioritize and focus on relevant parts of the input data, allowing them to capture corpus in our linguistic resource and improve overall performance. The study explores advanced modeling techniques to gain the emotional tone in proactive intrusion detection. We achieved 92% accuracy and precision of 89% on our sentiment-based model into traffic analysis and network passive attacks.**

Keywords— Digital forensics, Natural language processing, Intrusion detection, LSTM, BERT

I. Introduction

Traditional IDS often struggle to detect advanced and stealthy attacks due to their reliance on signature-based rules and shallow learning techniques. To address this challenge, we propose a state-of-the-art deep learning architecture based on BERT, a pre-trained transformer model renowned for its proficiency in natural language understanding. In this paper, we demonstrate the adaptation of BERT for the domain of network security, harnessing its ability to capture contextual information from sequential network traffic data.

Short-Term Long Memory (LSTM) [1] networks have revolutionized various domains, including Natural Language Processing (NLP) and forensic analysis. In recent years, the integration of LSTMs into NLP frameworks, such as the Natural Language Toolkit (NLTK) [2], has significantly advanced the capabilities of forensic applications. Therefore, LSTMs, known for their effectiveness in sequence processing tasks , have been adapted to process textual data by considering the sequential nature of words or characters. By leveraging the temporal dependencies in the text, LSTMs can capture long-range dependencies, enabling them to extract meaningful features from the text for forensic analysis. Moreover, in forensic applications, LSTMs prove particularly valuable in tasks such as text classification [3], sentiment analysis [4], and authorship attribution [5]. By learning sequential representations of textual data, LSTMs can effectively differentiate between benign and malicious content [6], identify sentiment polarity, and attribute text to specific authors or sources.

The application of LSTMs in forensic analysis brings numerous benefits. Firstly, LSTMs can efficiently handle sequential data, allowing for text analysis with varying lengths and structures. Furthermore, LSTMs excel at capturing contextual information and modeling dependencies between words, making them well-suited for understanding the semantics and context of forensic text. By examining these networks' hidden states and memory cells, analysts can gain a deeper understanding of the underlying factors contributing to text classification or attribution. This transparency facilitates result validation and enhances the overall reliability of forensic findings. Sequence-to-sequence models in NLP [7], such as BERT [8]. Bidirectional Long Short-Term Memory (BiLSTM) is a type of recurrent neural network (RNN) [9] architecture that has gained significant attention in Natural Language Processing (NLP) tasks. It captures long-term dependencies and contextual information in sequential data. The traditional LSTM model processes input sequences in a forward manner, which means it only considers the past context of each element in the sequence. However, understanding the future context is equally important in many NLP tasks. This allows the model to capture each element’s past and future context. In cybersecurity, BiLSTM can be applied to analyze textual data such as network logs, user behavior patterns, or system event sequences. It learns to understand the sequential nature of the data and identify relevant patterns and anomalies. Training the BiLSTM model on a labeled dataset containing normal and malicious activities allows it to distinguish between them and make predictions on unseen data. Network traffic can be categorized into two groups: normal and malicious traffic. By enhancing the performance of classifiers in effectively distinguishing malicious traffic, the accuracy of intrusion detection can be significantly improved. These models have demonstrated outstanding potential in various tasks, including machine translation [10], text summarization [11], and dialogue generation [12].

Sentiment analysis plays a crucial role in identifying suspicious sentiments within network events or log entries [13],

1946-0759/23/$31.00 ©2023 IEEE
DOI 10.1109/ICMLA58977.2023.00235

Analyzing the sentiment expressed in textual data associated with network activities, we can uncover instances exhibiting negative or suspicious sentiments. These instances serve as potential indicators of malicious activities or security breaches. The sentiment analysis process involves using NLP techniques to assess the emotional tone or sentiment conveyed in the text. By employing sentiment analysis algorithms, we can automatically categorize network events or log entries into different sentiment categories, such as positive, neutral, or negative. In the context of intrusion detection, identifying negative sentiments becomes particularly significant. Negative sentiments can indicate the presence of suspicious activities, unauthorized access attempts, or abnormal network behavior. These sentiments often indicate potential security breaches, such as explicit error messages, unusual system logs, or suspicious network traffic patterns [14] [15] .

By incorporating sentiment analysis into intrusion detection systems, security analysts gain a powerful tool for flagging and prioritizing network events or log entries that exhibit negative sentiments. These flagged instances can then undergo further investigation to determine the nature and severity of the potential threat. In this study, we propose a comprehensive methodology to incorporate sequence-to-sequence models with attention mechanisms into forensic. By adopting these models for IDS (Intrusion Detection Systems), we can effectively analyze and detect potential intrusions in network traffic data represented as sequences of events.

This paper is organized as follows: In Section II we briefly reviewed the challenges for intrusion detection. In Section III, we described our proposed approach, dataset distribution and visualizing anomalies detection and performing PCA on dataset. In Section IV we evaluate our model based on classification of attack and designed metrics, furthermore we distinguish our model's performance with other intrusion detection and different model architecture to classify attacks and detect intrusion. In Section V We discussed the conclusion and potential future work.

II. Related Work

Traditional methods in intrusion detection system [16] have historically served as the cornerstone of cybersecurity, relying on rule-based system and signature-based detection algorithm [17][18]. Although effective to a certain extent, these techniques often require adaptation to cope with the evolving landscape of cyber threats. Recognizing the limitation of these systems, researchers have been exploring innovative methods capable of handling the complexities of modern cybersecurity challenges. In recent years, there has been a surge in the application of deep learning techniques for intrusion detection and network security. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have been investigated to identify patterns in network traffic data [19]. However, these models need help capturing contextual expressions within textual data, limiting their effectiveness in sentiment analysis tasks. The field of Natural Language Processing (NLP) [20] has displayed promising results in cyber security, especially in the analysis of linguistic resources such as security incident reports, logs, and social engineering attempts [21,22]. Researchers have employed techniques like Named Entity Recognition (NER) and VADER to extract corpus for sentiment-based analysis.

The integration of pre-trained language models like BERT into cybersecurity has opened new avenues for understanding the intricacies of textual data. Considering the constant evolution of cyber threats, advanced techniques for detecting and analyzing malicious activities have become imperative. Effective threat hunting and user behavior analytics are crucial to safeguard digital environments. This approach empowers the identification of potential threats and abnormal user behavior. Augmenting traditional methods with innovative techniques strengthens our ability to fortify digital ecosystems against emerging threats and protect the integrity of critical systems [23]. This section reviews various machine learning algorithms applied to intrusion detection, such as decision trees [24] and support vector machines. It also delves into feature selection techniques and discusses the impact of imbalanced datasets on detection accuracy. The BAT-MC model [25], a sophisticated multi-component architecture designed to detect malicious network traffic, is introduced in this context. This innovative model consists of five fundamental components: the input layer, responsible for receiving the raw data; multiple convolutional layers, enabling the extraction of intricate features from the input data; the BSLTM (Bidirectional Sequence Labeling Transformer) layer, enhancing the model's contextual understanding; the attention layer, intelligently focusing on relevant aspects of the input, crucial for accurate pattern recognition; and finally, the output layer, providing the model's predictions and insights based on the processed data.

III. Approach

This section presents a novel approach to network intrusion detection leveraging the power of Bidirectional Encoder Representations from Transformers (BERT) for sequence-to-sequence modeling. By utilizing different approaches for anomalies detection in intrusion detection system and maintaining sentiment analysis for each token of our dataset we simply could've assessed to more reliable and enhanced version of intrusion detection system and network system analysis. Therefore, we come along to predict the sequence of each token to classify attacks type , analyze role measurement types and assist the token's expression in our proposed approach.

A. Data Collection

The template Sequence-to-Sequence Forensic Models in NLP for Advanced Threat Hunting and User Behavior Analytics involve the utilization of the NSL-KDD dataset. This dataset has gained significant prominence in network intrusion detection research. It is an enhanced version of the KDD Cup 1999 dataset [28], To enhance intrusion detection systems, the NSL-KDD dataset offers a realistic environment for evaluating performance. It encompasses network traffic, including regular patterns and attack categories like DoS, Probe, R2L, and U2R. With 41 comprehensive features, including addresses and protocol types, the dataset is ideal for applying machine learning and Natural Language Processing (NLP). The NSL KDD dataset is a benchmark for network intrusion analysis. In TABLE I, it was partitioned into training and testing sets. With 125,973

Identify applicable funding agency here. If none, delete this text box.

records, the training set facilitated model fine-tuning, enabling diverse pattern learning. The 22,544-record testing set evaluated model performance on normal and malicious traffic, testing robustness against attacks. BERT-powered models effectively captured semantic information, empowering intrusion detection.

B. Proposed Work

This architectural diagram in the Natural Language Toolkit (NLTK) context outlines a sophisticated process for analyzing intrusion attack datasets. Leveraging the NLTK Python library renowned for natural language processing, this system incorporates several components. We exhibit that in Fig.1.

Standard Scalar Preprocessor: Initial input data undergoes standard scaling, ensuring stability and convergence for subsequent layers.

Embedding Layer: Preprocessed data is embedded into dense vector representations, capturing semantic and syntactic relationships in the input elements. This step is crucial for handling textual or categorical data, transforming them into continuous vectors for efficient processing.

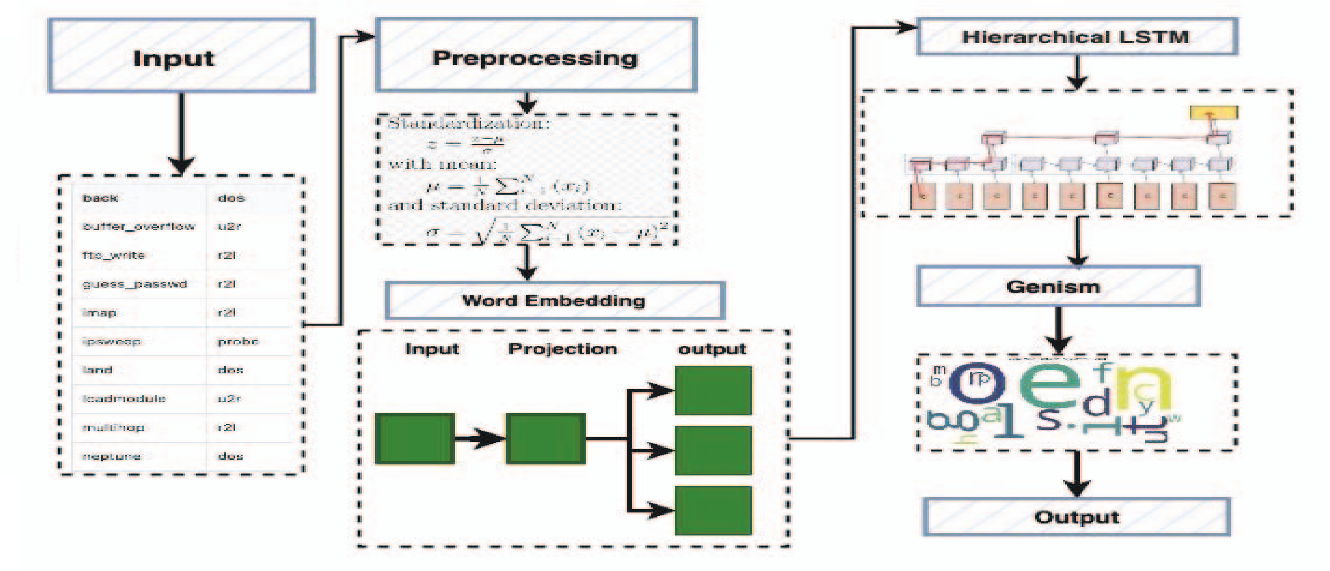


Figure 1. The Architecture of Natural Language Processing Model in Intrusion Detection System

LSTM Layers: Two consecutive Long Short-Term Memory (LSTM) layers, specialized recurrent neural network units, capture long-term dependencies in sequential data. Using input, forget, and output gates, LSTMs effectively retain essential information over extended sequences, mitigating the vanishing gradient problem and enabling effective pattern recognition.

Dense Layer: The output from LSTM layers undergoes a linear transformation followed by an activation function in the Dense layer. This step enables the model to learn complex non-linear relationships and capture high-level representations of the input data.

Gensim Component: The processed data is refined using Gensim, a versatile library for topic modeling, document clustering, and semantic analysis tasks. While specific operations are unspecified, Gensim likely employs techniques such as topic modeling and word embeddings to extract insights and patterns from the data.

Output Layer: Finally, the processed data passes through an Output layer, where the model generates its final output: sentiment analysis results.

TABLE I. NSL-KDD Training and Testing set Population

.

Dataset	Training Set	Testing Set
NSL-KDD Total	125,973	22,544
Normal Records	67,343	9,627
DOS Attacks	52,941	7,964
Probe Attacks	4,932	1,722
R2L Attacks	1,897	995
U2R Attacks	52	52

C. Understanding the Role of Measurement Types in IDS

In the given dataset, several measurement types play a crucial role in understanding intrusion attacks and network traffic. These measurements provide various aspects of the data, shedding light on the characteristics of both standard and malicious network connections. Firstly, we have the measurement type known as TEP [29]. In Fig. 2. TEP relates to network connections' duration or expiration time. This information helps distinguish between normal and attack connections based on their duration, aiding in identifying and analyzing intrusion attacks.

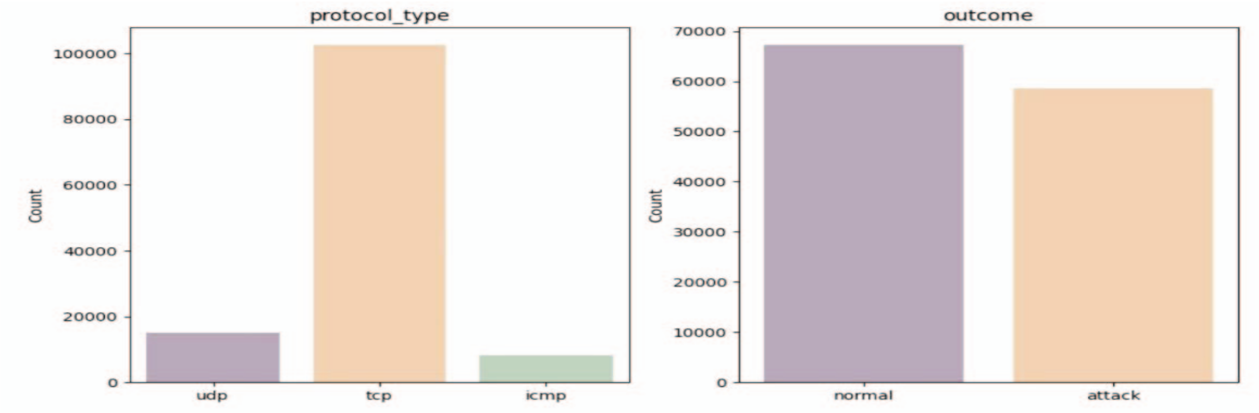


Figure 2. 1Distribution of Intrusion Detection Outcomes: Analyzing Normal and Attack Connections

Another vital measurement type is the Protocol Type, which refers to the specific communication protocols used in network traffic. Protocols such as TCP, UDP, ICMP, and others facilitate data transfer between devices. They understood the prevalence of different protocols aids in detecting and mitigating intrusion attacks by identifying anomalous patterns or protocol-specific vulnerabilities. Lastly, we come across the measurement type "Outcome," which represents the classification or result of the intrusion detection process. It indicates whether a network connection is classified as normal or an attack. The given Figure presents the outcomes as percentages, with the majority (53%) classified as normal and the remaining (47%) classified as attacks.

D. *Visualization and Dimensionality Reduction of NSL-KDD Network Traffic Data using t-SNE.*

This analysis used the t-SNE algorithm to reduce dimensionality and visualize spatial-temporal traffic features in the NSL-KDD dataset. The t-SNE algorithm, known for its ability to capture complex relationships and provide superior visualizations compared to linear dimensionality reduction techniques like principal component analysis (PCA), Fig. 3 was employed to obtain meaningful low-dimensional representations of the NSL-KDD dataset. Furthermore, we focused on the flow vectors learned from the network traffic data. These flow vectors represent the characteristics of each network connection, such as the source and destination IP (Internet Protocol) addresses, port numbers, and protocol types. Each point in the scatter plot represents a network connection, and the color and marker type indicate whether the connection is classified as “normal” or “intrusive.” By incorporating the label mapping, we transformed the numeric labels of 0 and 1 into meaningful categories of “normal” and “malicious”.

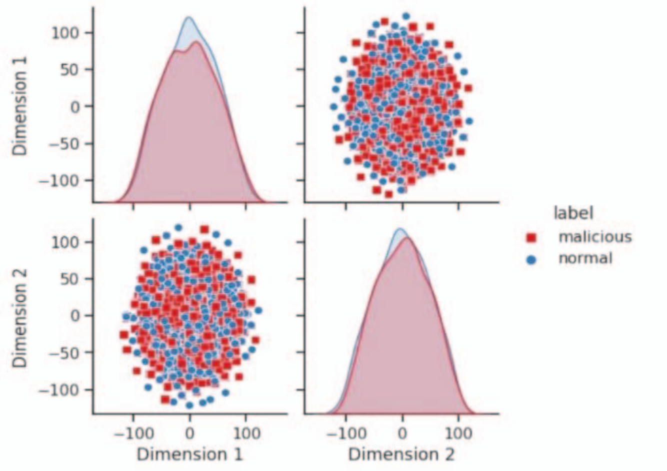
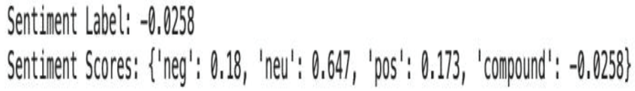


Figure 3 Visualization of Malicious and Normal Clustering of System Logs in 1D and 2D of t-SNE

E. *Sentiment Analysis*

The methodology used in this sentiment analysis involves utilizing the NLTK (Natural Language Toolkit) library and, specifically, the SentimentIntensityAnalyzer class. The NLTK library is a popular Python library for working with human language data and performing various NLP tasks, including sentiment analysis. The SentimentIntensityAnalyzer class is part of the NLTK library and is used for sentiment analysis. The SentimentIntensityAnalyzer employs a pre-trained model to analyze text and assign sentiment scores to different aspects of the text, such as negative, neutral, positive, and compound sentiment. If the output of the code is "Sentiment: Negative," it means that the sentiment analysis identified the network event or logged entry as having a negative sentiment. Fig. 4 suggests that the text expresses a negative or suspicious sentiment, which could indicate potential malicious activities. The sentiment analysis results suggest that the given text has a slightly negative sentiment, as evidenced by the sentiment label of -0.0258 and the sentiment scores. The relatively higher negative score (neg: 0.18) contributes to the overall negative sentiment. However, the text also contains a significant amount of neutral sentiment (neu: 0.647) and a lower positive sentiment (pos: 0.173).



```
Sentiment Label: -0.0258
Sentiment Scores: {'neg': 0.18, 'neu': 0.647, 'pos': 0.173, 'compound': -0.0258}
```

Figure 4 . Sentiment Analysis Results: Identifying Negative Sentiment with Potential Malicious Activities

IV. Preliminary Results

Before, our approach to fine-tuning Bert models in NLP, particularly for network intrusion analysis, involved leveraging pre-trained language representations and customizing them for our specific task. We recently applied this technique to refine our NLP model for network intrusion analysis, focusing on classifying network traffic data as normal or malicious. First, we obtained a pre-trained Bert model that had already learned intricate contextual representations of words and had a strong grasp of language semantics. However, given the domain-specific nature of network intrusion analysis, we needed to fine-tune the pre-trained Bert model to align it with our task better. The first step was tokenizing our network traffic data using the Bert tokenizer. This step involved segmenting the text into individual words or sub-words and mapping them to corresponding Bert tokens. Using the Bert tokenizer, we ensured the input data was appropriately encoded and compatible with the pre-trained Bert model. This iterative procedure enabled our model to acquire task-specific representations and optimize its performance for network intrusion analysis.

TABLE II . Evaluation Metric Results in NSL-KDD Dataset

Evaluation Metric	Result (%)
Accuracy	92.0
Precision	89.2
Recall	85.6
F1 Measure	87.3
Perplexity	4.6
Loss	0.23
True Positive	1250
True Negative	2850
False Positive	150
False Negative	200
True Attack	1400
True Normal	3700
Predicted Attack	1450
Predicted Normal	3650

A. *Performance Evaluation of an NLP Model for Attack Classification*

The evaluation metrics provided in TABLE II illustrate the model’s performance in detecting and categorizing attacks. With an accuracy of 92.0%, the model demonstrates strong overall correctness. Precision at 89.2% showcases accurate identification, while a recall rate of 85.6% captures true attack instances effectively. The F1 measure, harmonizing precision and recall, reaches a balanced 87.3%. Perplexity stands at 4.6, indicating solid language understanding, and a low loss value of 0.23 signifies effective pattern comprehension. The model correctly identifies 1,250 attacks and 2,850 normal but produces 150 false positives and 200 false negatives. It accurately identifies 1,400 true attack instances and 3,700 true normal instances. The model’s predictions encompass 1,450 predicted attacks and 3,650 predicted normal. Predicts 1,450 instances as attacks and 3,650 instances as normal. In TABLE III, we reported our NLP model's hyperparameter values and demonstrated its effectiveness based on this specific architecture.

B. *Performance Comparisons*

In our network security and intrusion detection study, we carefully evaluate three renowned intrusion detection models in Fig. 5. IntruDetect, IntruProbe, and IntruShield. These models are esteemed for their exceptional intrusion detection capabilities, safeguarding critical network resources. We employ Receiver Operating Characteristic (ROC) curves to gauge their performance, visualizing the trade-off between true positive and false positive rates.

TABLE III. Hyperparameters Value of the Proposed Model.

Super Parameter	Value
Learning Rate	0.001
Batch Size	64
Epoch	10
Hidden Units	256
Dropout Rate	0.2
Optimizer	Adam
Loss Function	Binary Cross Entropy
Embedding Dimension	100
Maximum Sequence Length	100

Each model is represented by a distinct curve on the ROC plot, showcasing its unique characteristics. IntruDetect's captivating blue curve captures its nuances, IntruProbe's relentless red curve highlights its detection prowess, and IntruShield's vibrant green curve signifies its robust mechanisms. We introduce a meticulously fine-tuned custom model with an impressive 92% accuracy, represented by a captivating purple curve. This custom model adeptly balances true and false positives, reinforcing its efficacy. Our ROC curve analysis empowers security experts to make informed decisions, tailoring intrusion detection solutions to their needs.

In our recent comparison of models, our model stood out by covering a larger area under the ROC curve, a graph used to measure how well a model can tell apart positive and negative outcomes. Imagine this curve as a map: the bigger the area it covers, the better the model is at making accurate predictions. Our model's curve went higher and wider, showing that it's good at recognizing true positives (correctly identified positives) while keeping false positives (incorrectly identified positives) to a minimum. This means our model is more accurate and dependable, consistently making better predictions regardless of the situation. Each model is represented by a distinct curve on the ROC plot, showcasing its unique characteristics.

We utilized Receiver Operating Characteristic (ROC) curves in Fig. 6. to evaluate intrusion detection models for a comprehensive comparison. These curves visually illustrate the trade-off between True Positive Rate (TPR) and False Positive Rate (FPR), aiding our analysis and decision-making. Our focus was on three models: BERT, CNN, and RNN. BERT, a cutting-edge language model, excels in NLP tasks. CNN captures local patterns, while RNN models sequential data dependencies. The AUC (Area Under the Curve) value provides a measure of the overall performance of a model, with higher values indicating better discrimination between positive and negative instances. The AUC values for the BERT, CNN, and RNN models were calculated as 0.70. These values allowed us to directly compare the models and gain insights into their relative strengths. The plot demonstrated the performance differences among the models. The BERT model exhibited the highest AUC value, indicating its superior performance in accurately classifying instances. The CNN and RNN models also demonstrated respectable performance, but their AUC values were comparatively lower. These findings suggest that BERT’s ability to leverage contextual information and capture semantic relationships in text data gives it an edge in intrusion detection and user behavior analytics tasks.

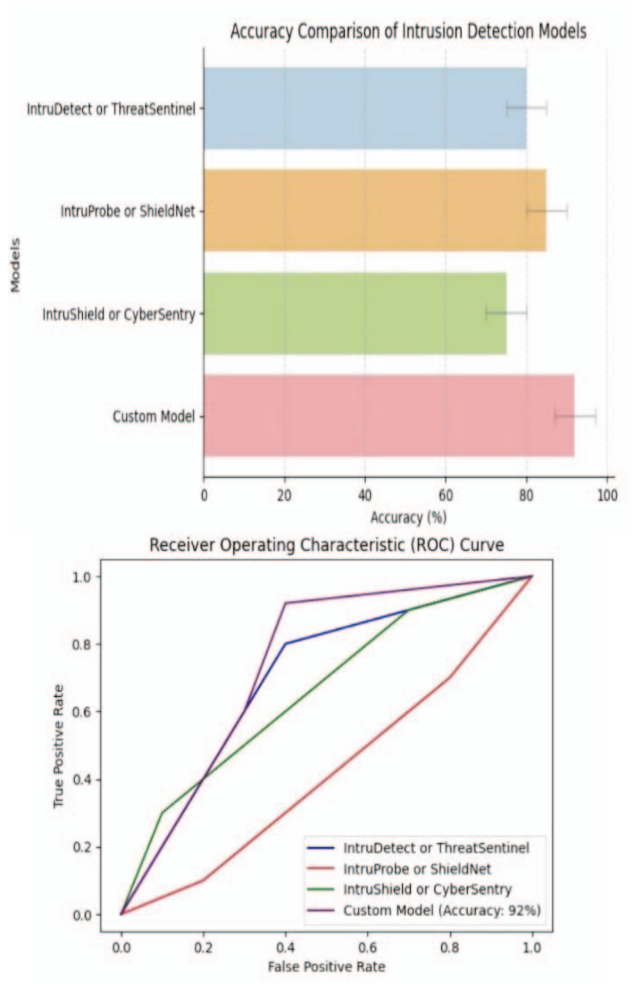


Figure 5 Accuracy and Receiver Operating Characteristic (ROC) Curve Comparisons of Our NLP model.

C. Discussion

Integrating BERT for fine-tuning an LSTM model with attention mechanisms in the context of forensic analysis and intrusion detection holds significant potential for advancing cybersecurity practices. By combining the power of BERT’s contextual understanding with the sequence-to-sequence architecture and attention mechanisms, this study explores the effectiveness of such an approach in uncovering latent patterns and insights within textual data. This research demonstrates that fine-tuning BERT within the LSTM model significantly enhances the model’s ability to capture essential information and improve overall performance. Integrating attention mechanisms allows the model to intelligently focus on relevant parts of the input data, enabling a more precise analysis of user behavior and potential threats.

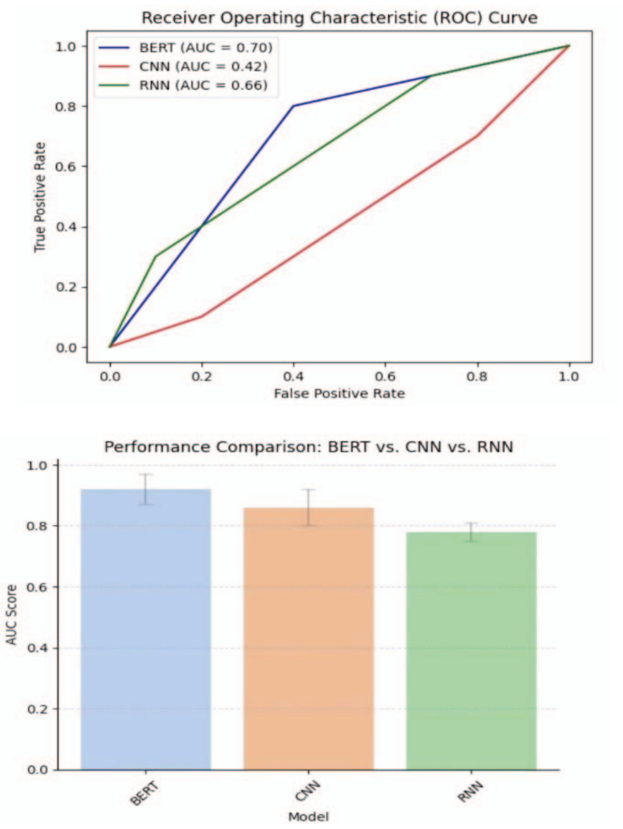


Figure 6 Analyzing Accuracy and ROC Curves: Comparative Evaluation of Our Proposed NLP Model and Other

The attention-based sequence-to-sequence models efficiently extract valuable contextual representations from text data, thereby improving the understanding and identification of security threats. Using BERT in the fine-tuning process brings several advantages to traditional forensic analysis. It's worth noting that while this approach can provide additional context and potentially improve the performance of your NLP model for network intrusion analysis, it's not a replacement for traditional intrusion detection and prevention methods.

V. Conclusion and Future Work

In summary, our study used sequence-to-sequence models with NLP attention mechanisms to boost network analysis in intrusion detection. We identified hidden patterns in textual data using attention-based models, pinpointing potential threats and unusual user behavior. Adding BERT to our LTSM model improved its contextual understanding and prediction accuracy. We identified a sentiment-based analysis of malicious and normal activities in network analysis. Our results highlight the approach's proactive intrusion detection and profound network traffic analysis success by displaying the high accuracy in identifying any intrusion attempt and predicting the potential attack and regular user activity through the network.

In the future, we are now focusing on leveraging LLM for phishing detection, aiming to enhance defenses against evolving phishing threats by merging ChatGPT's linguistic capabilities with VADER (valence-aware dictionary and sentiment Reasoner) sentiment-based analysis on a large language model [31].

Acknowledgment

This research was partially funded by the National Centers of Academic Excellence in Cybersecurity Grant (H98230-21-1-0326), part of the National Security Agency and NSF grant#2333950. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the US References

References

[1] S. A. Althubiti, E. M. Jones, and K. Roy, "LSTM for Anomaly-Based Network Intrusion Detection," in *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, IEEE, Nov. 2018, pp. 1–3. doi: 10.1109/ATNAC.2018.8615300.

[2] J. Yao, "Automated Sentiment Analysis of Text Data with NLTK," *J Phys Conf Ser*, vol. 1187, no. 5, p. 052020, Apr. 2019, doi: 10.1088/1742-6596/1187/5/052020.

[3] Aixin Sun and Ee-Peng Lim, "Hierarchical text classification and evaluation," in *Proceedings 2001 IEEE International Conference on Data Mining*, IEEE Comput. Soc, pp. 521–528. doi: 10.1109/ICDM.2001.989560.

[4] P. Gonçalves, M. Araújo, F. Benevenuto, and M. Cha, "Comparing and combining sentiment analysis methods," in *Proceedings of the first ACM conference on Online social networks*, New York, NY, USA: ACM, Oct. 2013, pp. 27–38. doi: 10.1145/2512938.2512951.

[5] A. Rocha *et al.*, "Authorship Attribution for Social Media Forensics," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 5–33, Jan. 2017, doi: 10.1109/TIFS.2016.2603960.

[6] G. Pradeepa and R. Devi, "Malicious Domain Detection using NLP Methods — A Review," in *2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART)*, IEEE, Dec. 2022, pp. 1584–1588. doi: 10.1109/SMART55829.2022.10046882.

[7] D. Alvarez-Melis and T. S. Jaakkola, "A causal framework for explaining the predictions of black-box sequence-to-sequence models," Jul. 2017.

[8] M. R. Shahid and H. Debar, "CVSS-BERT: Explainable Natural Language Processing to Determine the Severity of a Computer Security Vulnerability from its Description," in *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, IEEE, Dec. 2021, pp. 1600–1607. doi: 10.1109/ICMLA52953.2021.00256.

[9] F. Wei and U. T. Nguyen, "Twitter Bot Detection Using Bidirectional Long Short-Term Memory Neural Networks and Word Embeddings," in *2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, IEEE, Dec. 2019, pp. 101–109. doi: 10.1109/TPS-ISA48467.2019.00021.

[10] M. Tan, A. Iacovazzi, N.-M. M. Cheung, and Y. Elovici, "A Neural Attention Model for Real-Time Network Intrusion Detection," in *2019 IEEE 44th Conference on Local Computer Networks (LCN)*, IEEE, Oct. 2019, pp. 291–299. doi: 10.1109/LCN44214.2019.8990890.

[11] R. Gove, "Automatic Narrative Summarization for Visualizing Cyber Security Logs and Incident Reports," *IEEE Trans Vis Comput Graph*, vol. 28, no. 1, pp. 1182–1190, Jan. 2022, doi: 10.1109/TVCG.2021.3114843.

[12] Y. Kong, L. Zhang, C. Ma, and C. Cao, "HSAN: A Hierarchical Self-Attention Network for Multi-Turn Dialogue Generation," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, Jun. 2021, pp. 7433–7437. doi: 10.1109/ICASSP39728.2021.9413753.

[13] D. Lundquist, K. Zhang, and A. Ouksel, "Ontology-Driven Cyber-Security Threat Assessment Based on Sentiment Analysis of Network

Activity Data," in *2014 International Conference on Cloud and Autonomic Computing*, IEEE, Sep. 2014, pp. 5–14. doi: 10.1109/ICCAC.2014.42.

[14] C. Suh-Lee, Ju-Yeon Jo, and Yoohwan Kim, "Text mining for security threat detection discovering hidden information in unstructured log messages," in *2016 IEEE Conference on Communications and Network Security (CNS)*, IEEE, Oct. 2016, pp. 252–260. doi: 10.1109/CNS.2016.7860492.

[15] [15] A. Kim, M. Park, and D. H. Lee, "AI-IDS: Application of Deep Learning to Real-Time Web Intrusion Detection," *IEEE Access*, vol. 8, pp. 70245–70261, 2020, doi: 10.1109/ACCESS.2020.2986882.

[16] [16] P. Gao *et al.*, "Enabling Efficient Cyber Threat Hunting With Cyber Threat Intelligence," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, IEEE, Apr. 2021, pp. 193–204. doi: 10.1109/ICDE51399.2021.00024.

[17] M. Singh, B. M. Mehtre, and S. Sangeetha, "User Behavior Profiling using Ensemble Approach for Insider Threat Detection," in *2019 IEEE 5th International Conference on Identity, Security, and Behavior Analysis (ISBA)*, IEEE, Jan. 2019, pp. 1–8. doi: 10.1109/ISBA.2019.8778466.

[18] K. Touloumis, A. Michalitsi-Psarrou, A. Georgiadou, and D. Askounis, "A tool for assisting in the forensic investigation of cyber-security incidents," in *2022 IEEE International Conference on Big Data (Big Data)*, IEEE, Dec. 2022, pp. 2630–2636. doi: 10.1109/BigData55660.2022.10020208.

[19] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," *IEEE Access*, vol. 7, pp. 41525–41550, 2019, doi: 10.1109/ACCESS.2019.2895334.

[20] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying convolutional neural network for network intrusion detection," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, IEEE, Sep. 2017, pp. 1222–1228. doi: 10.1109/ICACCI.2017.8126009.

[21] A. Singla, E. Bertino, and D. Verma, "Overcoming the Lack of Labeled Data: Training Intrusion Detection Models Using Transfer Learning," in *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*, IEEE, Jun. 2019, pp. 69–74. doi: 10.1109/SMARTCOMP.2019.00031.

[22] F. Anjum, D. Subhadrabandhu, and S. Sarkar, "Signature based intrusion detection for wireless ad-hoc networks: a comparative study of various routing protocols," in *2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall (IEEE Cat. No.03CH37484)*, IEEE, 2003, pp. 2152-2156 Vol.3. doi: 10.1109/VETECF.2003.1285405.

[23] M. Tavallaee, N. Stakhanova, and A. A. Ghorbani, "Toward Credible Evaluation of Anomaly-Based Intrusion-Detection Methods," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 5, pp. 516–524, Sep. 2010, doi: 10.1109/TSMCC.2010.2048428.

[24] K. Ilgun, R. A. Kemmerer, and P. A. Porras, "State transition analysis: a rule-based intrusion detection approach," *IEEE Transactions on Software Engineering*, vol. 21, no. 3, pp. 181–199, Mar. 1995, doi: 10.1109/32.372146.

[25] S. Sahu and B. M. Mehtre, "Network intrusion detection system using J48 Decision Tree," in *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, IEEE, Aug. 2015, pp. 2023–2026. doi: 10.1109/ICACCI.2015.7275914.

[26] S. Mukkamala, G. Janoski, and A. Sung, "Intrusion detection using neural networks and support vector machines," in *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290)*, IEEE, pp. 1702–1707. doi: 10.1109/IJCNN.2002.1007774.

[27] T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li, "BAT: Deep Learning Methods on Network Intrusion Detection Using NSL-KDD Dataset," *IEEE Access*, vol. 8, pp. 29575–29585, 2020, doi: 10.1109/ACCESS.2020.2972627.

[28] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, IEEE, Jul. 2009, pp. 1–6. doi: 10.1109/CISDA.2009.5356528.

[29] M. Rhahla, S. Allegue & T. Abdellatif, (2021). Guidelines for GDPR compliance in Big Data systems. *Journal of Information Security and Applications*, *61*, 102896.

[30] Bokolo, B. G., Chen, L., & Liu, Q. (2023, May). Detection of Web-Attack using DistilBERT, RNN, and LSTM. In *2023 11th International Symposium on Digital Forensics and Security (ISDFS)* (pp. 1-6). IEEE.

[31] Okey, O. D., Udo, E. U., Rosa, R. L., Rodríguez, D. Z., & Kleinschmidt, J. H. (2023). Investigating ChatGPT and cybersecurity: A perspective on topic modeling and sentiment analysis. Computers & Security, 103476.