

The Application of Transformer-Based Models for Predicting Consequences of Cyber Attacks

Bipin Chhetri and Akbar Siami Namin

Department of Computer Science

Texas Tech University

Lubbock, TX, USA

bipin.chhetri@ttu.edu, akbar.namin@ttu.edu

This work has been published in COMPSAC Symposium on Emerging Advances in Technologies & Applications (EATA 2025), IEEE COMPSAC 2025 IEEE International Conference on Computers, Software, & Applications, Toronto, Canada, July 8-11, 2025

Abstract

Cyberattacks are increasing, and securing against such threats is costing industries billions of dollars annually. Threat Modeling, that is, comprehending the consequences of these attacks, can provide critical support to cybersecurity professionals, enabling them to take timely action and allocate resources that could be used elsewhere. Cybersecurity is heavily dependent on threat modeling, as it assists security experts in assessing and mitigating risks related to identifying vulnerabilities and threats. Recently, there has been a pressing need for automated methods to assess attack descriptions and forecast the future consequences of the increasing complexity of cyberattacks. This study examines how Natural Language Processing (NLP) and deep learning can be applied to analyze the potential impact of cyberattacks by leveraging textual descriptions from the MITRE Common Weakness Enumeration (CWE) database. We emphasize classifying attack consequences into five principal categories: Availability, Access Control, Confidentiality, Integrity, and Other. This paper investigates the use of Bidirectional Encoder Representations from Transformers (BERT) in combination with Hierarchical Attention Networks (HANs) for Multi-label classification, evaluating their performance in comparison with conventional CNN and LSTM-based models. Experimental findings show that BERT achieves an overall accuracy of 0.972, far higher than conventional deep learning models in multi-label classification. HAN outperforms baseline forms of CNN and LSTM-based models on specific cybersecurity labels. However, BERT consistently achieves better precision and recall, making it more suitable for predicting the consequences of a cyberattack.

Keywords: Hierarchical Attention Networks (HAN), Bidirectional Encoder Representations from Transformers (BERT), Long Short-Term Memory (LSTM), Consequences of Cyber Attacks, Threat Modeling.

1 Introduction

Cyberattacks are becoming increasingly frequent and sophisticated, affecting critical infrastructure, cloud services, and healthcare systems on an unprecedented scale. In August 2023, Amazon Web Services (AWS) experienced a significant Distributed Denial of Service (DDoS) attack, targeting Amazon S3 and resulting in more than 155 million requests per second. The incident caused an eight-hour outage, which affected access to vital services and highlighted the vulnerability of the cloud infrastructure to major cyber threats [3]. In another incident on that date, Google Cloud services experienced an alarming DDoS attack driven by HTTP/2, which peaked at 398 million requests per second. Unlike all previous events [18], this incident was the biggest Layer 7 assault known to date.

Beyond cloud services, the healthcare industry has also experienced numerous instances of cyberattacks. In February 2024, Change Healthcare [22] experienced a significant ransomware attack, compromising the medical and personal records of about 190 million people. The attack revealed private information, including names, phone numbers, social security numbers, and medical histories, resulting in one of the most significant health breaches, which caused significant operational challenges [22].

Threat modeling is an indispensable process within the field of cybersecurity, providing a structured methodology for security experts to assess and understand the potential threats and vulnerabilities inherent to complex systems [34]. The application of machine learning in cybersecurity has attracted a myriad of interests and is widely utilized to address security-related problems such as intrusion detection [6, 26, 29], malware analysis [2, 27], anomaly detection [1, 20] and vulnerability detection [4], [19]. Researchers continually devise new methods and strategies to enhance machine learning’s capabilities in tackling cybersecurity problems. However, the rapid advancement of technology has provided criminals with the opportunity to target frequent and sophisticated threats that were previously unattainable. The attacks on AWS [3] and Google [18] serve as a lesson of how modern cyber threats can be carried out with unprecedented speed and efficiency.

Cyberattacks are increasing in frequency and complexity. Therefore, protecting systems is now vital. By allowing for a more efficient and context-aware examination of whole text sequences [35], transformer models have greatly improved Natural Language Processing (NLP). Traditional models, such as Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs), consider an input in a specific order or focus on local features. On the other hand, transformers utilize self-attention methods to determine the importance of each word in a given phrase, and therefore have evolved into a fundamental component of modern NLP systems.

Inspired by a previous study by Datta et al. [10], where the application of CNN and LSTM models for multi-label classifications was explored to predict consequences of cyberattacks, this paper introduces a hybrid approach that combines these deep architectures with attention mechanisms and ensemble learning through transformer-based models such as BERT transformer models and Hierarchical Attention Networks (HAN) with the objective of enhancing the accuracy of predicting consequences of cyberattacks. The goal is to improve the accuracy of consequence prediction,

particularly for consequences modeled through mainstream threat modeling schema such as CIR (i.e., Confidentiality, Integrity, and Availability). This paper specifically focuses on predicting multiple labels, including Availability, Access Control, Confidentiality, Integrity, and Other associated with each textual description drawn from MITRE’s Common Weakness Enumeration (CWE) dataset. [28] The real-world results of our study demonstrate that both the Hierarchical Attention Networks (HANs) deep neural network model and the BERT transformer model outperform the state-of-the-art CNN and RNN-based models.

This study addresses the critical challenge of predicting the consequences of cyberattacks that have implications for end-users, a task complicated by the multi-label nature of cybersecurity vulnerabilities and the limitations of existing models in handling such complexity. Traditional methods, such as CNN, RNN, and knowledge graph-based approaches, often overlook the complex relationships in textual data, despite the models’ need to understand the broader context. By examining these issues, the study helps to enhance threat modeling techniques and improve the accuracy of predictions about potential outcomes following a cyberattack incident. The paper makes the following key contributions:

- **Transformer-based BERT multi-label Classification.** We introduce a transformer-based BERT model fine-tuned for predicting five key labels representing consequences of cyberattacks, such as Availability, Access Control, Confidentiality, Integrity, and Other, using textual descriptions from the MITRE CWE dataset [28].
- **Hierarchical Attention Networks (HAN).** The study demonstrates the effectiveness of HANs in capturing document-level semantics, enhancing the precision of multi-label classification for two labels, i.e., Access control and Integrity.
- **Improved Performance.** Empirical results show that the BERT-based model significantly outperforms traditional CNN-LSTM architectures, achieving superior accuracy and F1-scores across all labels. Notably, BERT achieves the best performance overall, while HAN excels in specific categories (i.e., two labels).
- **Practical and Scalable Solution.** This work offers a scalable alternative to knowledge graph-based methods, providing a streamlined approach suitable for real-world applications in cybersecurity.

The paper is organized as follows. Section 2 introduces the related work. Section 3 provides a brief overview of the technical methodologies adopted in this paper. Section 4 presents the architecture and implementation of the proposed models. Section 5 provides the details of the dataset preprocessing techniques and experimental setup. The methodology employed in this paper is described in Section 6. Section 7 discusses the analysis and results. Section 8 compares the proposed approach with previous work. Section 9 concludes the paper and outlines future research directions.

2 Related work

Various deep learning models, such as Long Short-Term Memory networks (LSTMs), Recurrent neural network (RNN), Convolutional Neural Networks (CNNs), and hybrid models have been extensively applied in cybersecurity for tasks like anomaly detection [36], Entity recognition [17] and threat mitigation [39]. These models often face challenges when analyzing complex textual data inherent in cybersecurity contexts [16]. With its self-attention methods, transformer-based models are known as a major development as they provide better performance on several NLP tasks, including text categorization. This section provides an overview of current deep learning techniques, Hybrid-LSTM and RNN, and transformer-based models that are initially used for various applications mitigating cyberattacks.

Hui et al. [24] proposed a fusion model that combines the attention mechanism with Bi-LSTM and BERT, which significantly outperforms single-model approaches. Their model achieved an accuracy of 89.52% on short medical text from Traditional Chinese Medicine (TCM) medical records, highlighting the potential of combining contextual embeddings with sequential learning. Their work is limited to short medical texts, which may not generalize well to longer, more complex documents. Additionally, their study does not address challenges in multi-label texts, which may not be generalized constraints such as cybersecurity.

Mahdaouy et al. [14] used a pre-trained BERT-based encoder designed for detecting and classifying suspicious or malicious domain names and URLs. For pre-training, the Masked Language Modeling (MLM) objective was applied to a large multilingual dataset that included URLs, domain names, and Domain Generation Algorithm (DGA) data. The authors evaluated the performance of their models on various classification tasks, including phishing, malware detection, and DNS tunneling. The result showed DomURLs_BERT outperforms state-of-the-art character-based deep learning models across multiple datasets. The study does not thoroughly address its vulnerability to adversarial attacks.

Cheng et al. [8] proposed a Hierarchical Attention Network in cyberbullying detection (HANCD) that captures temporal behavior patterns of cyberbullying detection. A real-world Instagram dataset from [23] demonstrates that incorporating temporal dynamics improves performance by 5.3% compared to the Hierarchical Attention Network with Temporal Features (HANT). The study is limited to Instagram, and the approach may not be readily generalized to other social media platforms, such as Facebook and X.

Xu et al. [38] introduced a dual-domain intrusion detection (DDT) model that combines Temporal Convolutional Networks (TCN) to extract local and global features, addressing the growing complexity of network attacks. Their work not only focuses on intrusion detection but also highlights the potential of Transformer-based models in cybersecurity applications. The DTT model exhibits improvements in F1-score ranging from 0.6 to 6.8 on the NCCI dataset and from 0.4 to 3.5 on the NUB dataset compared to other models. The model requires extensive pre-training, making it less suitable for real-time intrusion detection scenarios.

Nguyen et al. [30] proposed an approach to improve the domain adaptation capability of Network

Intrusion Detection Systems (NIDS) using Natural Language Processing (NLP) and the BERT framework. The network traffic flows were organized as sequences, similar to sentences in language. The authors trained the BERT model using the Masked Language Modeling (MLM) task and then fine-tuned it with a linear layer and softmax output for intrusion detection. It achieved an F1-score of 0.877 and an accuracy of 0.916. The model achieved positive results across different domains. The model still relies on labeled data, which limits its scalability and applicability.

Deep learning models have been extensively utilized for multi-label text classification tasks. Early works, such as Kim [25], employed a simple Convolutional Neural Network (CNN) with pre-trained word2vec embeddings to predict sentiment and classify questions. This study highlighted the effectiveness of CNNs in text classification and laid the groundwork for subsequent hybrid models. Later methods combined CNNs and Recurrent Neural Networks (RNNs) to leverage the strengths of both types of architecture.

In cybersecurity, knowledge graphs have emerged as a promising tool for modeling complex relationships among vulnerabilities, threats, and consequences. Han et al. [21] constructed a knowledge graph using CWE data to predict relationships between entities and classify cybersecurity consequences. These efforts have automated processes such as assigning CWE-IDs to CWE entries and have shown potential in organizing cybersecurity information more effectively. Their reliance on structured graph representations contrasts with this study, which leverages deep learning techniques to predict multi-label outcomes without the need for knowledge graphs.

Recent research has also explored machine learning techniques for predicting attack consequences in cybersecurity. Datta et al. [11] applied machine learning models to analyze 93 attack descriptions and predict 50 potential consequences from an end-user perspective. Similarly, Dass et al. [9] utilized Hidden Markov Models (HMMs) to anticipate attack consequences from the victim’s perspective, providing a proof-of-concept for spoofing attacks. These studies underscore the importance of accurately modeling and predicting the impacts of attacks, yet they rely on traditional techniques that may not scale effectively to larger datasets.

The work presented in this paper fills this gap by utilizing transformer-based models for multi-label classification, which allows us to predict cybersecurity attack outcomes exclusively from textual descriptions. Previous research has primarily employed knowledge graphs or basic machine learning techniques. This study, on the other hand, utilizes advanced attention and transformer-based architectures, such as BERT and HAN, to enhance the accuracy and scalability of multi-label classification in cybersecurity settings.

3 Preliminary

3.1 Transformers

The Transformer-based models, introduced by Vaswani et al. [35], have dramatically changed deep learning, as a self-attention mechanism is able to capture long-range dependencies without dealing with the inherent problems (e.g., vanishing gradient) that often occur in recurrent or convolutional

neural networks. Unlike RNNs and LSTMs, which process input sequentially, transformers are able to operate in parallel, resulting in huge efficiency and scalability benefits. The multi-head self-attention is used to give the model the chance to assess the relevance of different words in a sentence by itself. The model architecture utilizes "position encoding" to preserve word order information. This architecture has led to the most recent breakthroughs in multiple areas, such as natural language processing [5, 31], time series forecasting [37], and computer vision [7, 13], with models including BERT, GPT, and T5 demonstrating deeper contextual knowledge. Despite its impressive performance, the Transformer's high computational cost and memory usage severely limit its capacity. This is the driving ongoing research into the domain of more efficient attention techniques, such as sparse and linear attention mechanisms. As optimizations continue to improve in terms of scalability, transformers are at the heart of modern AI research and applications.

3.2 Hierarchical Attention Network (HAN)

Hierarchical Attention Networks (HANs) is an effective model used in document-level representations, particularly for long-text classification tasks [40]. Unlike conventional deep learning models, HAN leverages attention mechanisms at multiple levels words and sentences to capture contextual meaning while focusing on the most relevant parts of the text. Given the multi-label nature of cybersecurity vulnerability classification, HAN is a suitable choice because it can highlight informative contents from textual descriptions.

The architecture comprises word-level and sentence-level attention mechanisms that aggregate meaningful representations for classification. The model begins by processing tokenized words through an embedding layer, such as GloVe, followed by a bidirectional Gated Recurrent Unit (GRU) that generates context-aware word embeddings. A word-level attention mechanism assigns weights to words based on their importance in the classification task. The sentence-level attention follows a similar approach, where word representations are aggregated into sentence embeddings, processed again by a bidirectional GRU, and refined through attention to highlight significant sentences. The final classification layer maps the attended sentence representations to multiple labels using a fully connected dense layer with a softmax or sigmoid activation function. This hierarchical approach enables HAN to efficiently capture contextual relationships within text and improve classification accuracy.

4 Transformer-based Model Architectures for Predicting Consequences of Cyberattacks

4.1 The BERT Model

The foundation of this model is based on the pre-trained "*bert-base-uncased*" [15], which is derived from the BERT architecture initially proposed in [12]. This version is one of the first pre-trained models that was made public and fine-tuned for classification tasks where the output can be any

consequence (e.g., integrity, confidentiality, etc.). Because it can utilize contextual embeddings, it is a suitable choice for many natural language processing applications. A pre-trained **BERT** (Bidirectional Encoder Representations from Transformers) model (**bert-base-uncased**) was employed, comprising the following components:

1. **BERT Encoder.** The encoder comprises 12 transformer layers that generate contextually rich embeddings for input tokens, leveraging self-attention mechanisms to learn relationships between words within the text.
2. **Input Layer.** The model processes tokenized sequences derived from cybersecurity vulnerability descriptions, with a maximum sequence length set of 256 tokens.
3. **Dropout Layer.** Applied with a probability of 0.3 to prevent overfitting.
4. **Linear Layer.** Maps the 768-dimensional pooled output of the BERT model to five output neurons corresponding to the multi-label targets. A **sigmoid** activation function was applied to each of the five output neurons to predict each label’s probability independently. The sigmoid activation function is commonly used in binary classification tasks that require a value between 0 and 1, making it ideal for predicting the likelihood of an event occurring [32].



(a) BERT-Based Model Architecture

(b) HAN-Based Model Architecture

Figure 1: Architectural Diagrams of BERT and HAN Models Used for Predicting Cyberattack Consequences

Figure 1a illustrates the architecture of the BERT transformer model with multi-labeled classification capability to predict the consequences of cyber attacks.

4.2 Hyper-parameter Tuning for BERT

Hyper-parameter tuning was crucial in fine-tuning the model for optimal performance. The following key hyper-parameter tunings were applied to the BERT model:

Algorithm 1 BERT-Based Classification Training Process.

```
1: Input: Training Dataset
2: Output: Trained BERT model
3: Split Data: Training, Validation, Testing.
4: procedure TRAIN MODEL
5:   Initialize BERT model with pre-trained weights
6:   for each epoch  $e = 1$  to  $E$  do
7:     for each mini-batch  $B \in Training$  do
8:       Tokenize  $B$ 
9:       Apply dropout and fully connected layers
10:    end for
11:    Evaluate validation performance
12:  end for
13:  Return trained BERT model
14: end procedure
15: procedure PREDICT(Testing Data  $X_{test}$ )
16:   Tokenize  $X_{test}$ 
17:   Pass tokens through trained BERT model
18:   Apply sigmoid activation
19:   Convert scores to binary labels using a threshold
20:   Return predicted labels  $\hat{Y}$ 
21: end procedure
```

- **Maximum Sequence Length:** 256 tokens, ensuring that longer descriptions are fully captured without truncation.
- **Batch Size:** A batch size of 32 was utilized during both training and validation.
- **Learning Rate:** A learning rate of 1×10^{-5} , optimized using the Adam optimizer, was found to be the most effective after testing different values.

4.3 Algorithm for Predicting Consequences with BERT Layer

Algorithm 1 outlines the BERT-based training and prediction steps. Initially, the dataset is split into training, validation, and testing subsets to ensure fair evaluation. Mini-batches generated from the training data are used to fine-tune the model after it has been initialized with pre-trained weights. Each mini-batch undergoes tokenization using the BERT tokenizer, after which the tokenized input is fed through the encoder layers. A dropout regularization technique is then applied, followed by a fully connected output layer for multi-label classification. During training, the binary cross-entropy loss is calculated and backpropagated to update the model weights. In the prediction phase, the trained model feeds test data through the same pipeline to get probability distributions for all labels.

4.4 The HAN Model

The foundation of this model is a Hierarchical Attention Network (HAN), which is designed to capture document-level structures by leveraging both word-level and sentence-level attention mechanisms. Inspired by the work performed by Yang et al. [40], the HAN model is particularly effective for text classification tasks, as it learns to focus on the most informative words and sentences when making predictions. For this specific task, HAN was adapted for multi-label classification, predicting consequences such as Integrity, Confidentiality, Availability, Access Control, and Other. A Hierarchical Attention Network consists of the following key components:

1. **Word Encoder.** A bidirectional Gated Recurrent Unit (Bi-GRU) was employed to learn the contextual representation of each word within a sentence. The embedding layer converts tokenized words into dense vectors, which are then passed through the Bi-GRU.
2. **Word-Level Attention.** An attention mechanism is applied at the word level to assign weights to words based on their importance within the sentence. This allows the model to focus on key terms that contribute most to classification.
3. **Sentence Encoder.** The weighted word representations are aggregated into a sentence vector, which is then passed through another Bi-GRU layer to learn sentence dependencies.
4. **Sentence-Level Attention.** Similar to word-level attention, a sentence-level attention mechanism is employed to assess the significance of each sentence within the overall document, thereby enhancing classification accuracy.
5. **Fully Connected Layer.** The final document representation is passed through a fully connected layer with five output neurons, each representing a multi-label category.
6. **Sigmoid Activation.** A sigmoid activation function is applied to each of the five output neurons to produce independent probabilities for each consequence label.

Figure 1b illustrates the Hierarchical Attention Network architecture for predicting cyber attack consequences.

4.5 Hyper-parameter Tuning for HAN

Hyperparameter tuning was crucial for fine-tuning the HAN model to achieve optimal performance. The following key hyperparameters were used:

- **Maximum Sequence Length:** 256 tokens, ensuring adequate representation of cybersecurity vulnerability descriptions.
- **Batch Size:** A batch size of 32 was used for both training and validation.
- **Learning Rate:** 1×10^{-4} , optimized using the Adam optimizer.
- **GRU Units:** The GRU layers at the word level and sentence level consist of 64 hidden units.

Algorithm 2 HAN-Based multi-label Classification Training.

```
1: Input: Training Dataset
2: Output: Predicted labels for test data
3: Split Data: Training, Validation, Testing.
4: procedure TRAIN MODEL
5:   Initialize word and sentence Bi-GRU layers with attention mechanisms
6:   for each epoch  $e = 1$  to  $E$  do
7:     for each mini-batch  $B \in Training$  do
8:       Tokenize and embed words
9:       Encode words using Bi-GRU
10:      Apply attention layer
11:      Pass fully connected layer
12:    end for
13:    Validate Model performance
14:  end for
15: end procedure
```

4.6 Algorithm for Predicting Consequences with HAN Model

	description	Clean_Description
0	Information sent over a network can be compromised while in transit. An attacker may be able to read or modify the contents if the data are sent in plaintext or are weakly encrypted.	information sent network compromised transit attacker may able read modify contents data sent plaintext weakly encrypted
1	The J2EE application is configured to use an insufficient session ID length.	j2ee application configured use insufficient session id length
2	The default error page of a web application should not display sensitive information about the software system.	default error page web application display sensitive information software system
3	When an application exposes a remote interface for an entity bean, it might also expose methods that get or set the bean's data. These methods could be leveraged to read sensitive information, or to change data in ways that violate the application's expectations, potentially leading to other vulnerabilities.	application exposes remote interface entity bean might expose methods get set beans data methods could leveraged read sensitive information change data ways violate applications expectations potentially leading vulnerabilities
4	If elevated access rights are assigned to EJB methods, then an attacker can take advantage of the permissions to exploit the software system.	elevated access rights assigned ejb methods attacker advantage permissions exploit software system

Figure 2: A sample of CWE dataset with the original description and description after cleaning.

Algorithm 2 outlines the HAN-based training and prediction steps. The training corpus is first divided into stratified subsets to preserve the distribution of multi-label classes. The model is then initialized with Bi-GRU encoders at both the word and sentence levels, each followed by dedicated attention modules. Pre-trained word vectors are used to tokenize and embed input text during each training cycle. The word sequences in each sentence are passed through a word-level Bi-GRU, where an attention mechanism assigns weights to their contextual embeddings, prioritizing semantically relevant tokens. The aggregated weighted word embeddings were then used to construct sentence representations, which a sentence-level Bi-GRU processed.

5 Experimental Setup

5.1 Data Pre processing

The dataset used in this study is derived from an enhanced version of the MITRE Common Weakness Enumeration (CWE) dataset [28]. The CWE database is continuously updated and maintained

to reflect newly identified attacks. At the time of data collection, it contained 1,016 distinct CWE entries. The task is set up as a multi-label classification problem because we need to predict the outcome based on text descriptions, and each CWE entry can be linked to more than one outcome. Initially, there were six labels. However, to balance the dataset, we retained five columns: availability, access control, confidentiality, integrity, and non-repudiation, among others. To ensure a balanced distribution of the multi-label classes across the subsets, the dataset was divided using stratified sampling. After filtering the dataset, **895 rows** remained, each containing descriptions of cybersecurity vulnerabilities labeled with one or more consequences. The five target labels for this study were: **Availability**, **Access Control**, **Confidentiality**, **Integrity**, and **Other**.

5.2 Data Cleaning

The initial dataset includes raw text data, which may contain unnecessary symbols, stop words, and inconsistent capitalization. These artifacts needed to be cleaned for efficient processing. Figure 2 depicts a sample of the CWE dataset before preprocessing of the description and the cleaning stage. Once the necessary preprocessing techniques, such as removing stopwords, tokenization, and normalizing the text, was applied, the dataset was cleaned and made ready for model training. Prior to training, the following preprocessing steps were applied to the dataset:

- **Text Cleaning.** Descriptions were cleaned using Python-based NLTK (Natural Language Toolkit), removing irrelevant information such as punctuation, stopwords, and non-text symbols. The text was also converted to lowercase to standardize the input.
- **Target Labels.** Only five key target labels were retained (*Availability*, *Access Control*, *Confidentiality*, *Integrity*, *Other*), and redundant fields like *id*, *name*, and *extended_description* were excluded to focus the dataset on the most relevant information.

After the cleaning process and the removal of labels associated with smaller subsets, Figure 3 depicts a cleaned CWE description paired with its corresponding five multi-label classifications.

5.3 Tokenization

The *BertTokenizer* from the Hugging Face transformer library [15] was used to tokenize the text descriptions. The text was encoded into input tokens with padding and truncation applied to ensure a uniform sequence length $MAX_LEN = 256$.

5.4 Performance Metrics

Metrics such as **accuracy**, **precision**, **recall**, and **F1-score** were calculated for both micro and macro averages. These metrics were computed using the Scikit-learn library [33]. Validation was carried out at the end of each epoch to monitor performance. The model performance evaluation details are as follows:

	Clean_Description	Availability	Access_Control	Confidentiality	Integrity	Other
0	information sent network compromised transit attacker may able read modify contents data sent plaintext weakly encrypted	0	0	1	1	0
1	j2ee application configured use insufficient session id length	0	1	0	0	0
2	default error page web application display sensitive information software system	0	0	1	0	0
3	application exposes remote interface entity bean might expose methods get set beans data methods could leveraged read sensitive information change data ways violate applications expectations potentially leading vulnerabilities	0	0	1	1	0
4	elevated access rights assigned ejb methods attacker advantage permissions exploit software system	0	0	0	0	1
5	debugging messages help attackers learn system plan form attack	0	0	1	0	0
6	asp net application must enable custom error pages order prevent attackers mining information frameworks builtin responses	0	0	1	0	0
7	storing plaintext password configuration file allows anyone read file access passwordprotected resource making easy target attackers	0	1	0	0	0
8	sensitive memory cleared according source code compiler optimizations leave memory untouched read aka dead store removal	0	1	1	0	0

Figure 3: A sample of the CWE dataset with the clean description and five labels.

- **Accuracy:** The proportion of correctly classified labels over the total number of labels.
- **Precision:** The ratio of true positives to all predicted positives, relevant in cases of imbalanced data.
- **Recall:** The ratio of true positives to all actual positives, indicating the model’s ability to capture relevant instances.
- **F1-Score:** The harmonic mean of precision and recall, calculated for both micro and macro averages.

6 Methodology

6.1 Training Process

The training loop involved minimizing the binary cross-entropy loss between predicted and actual labels. Model checkpoints were saved at each epoch if the validation loss improved, and early stopping was employed to prevent overfitting. A validation set was used at the end of each epoch to monitor performance and prevent overfitting. The model was trained on an **NVIDIA A100 GPU** for accelerated processing.

6.2 Data Splitting

The *train_test_split* function from *Scikit-Learn* was used with the *stratify* parameter to split the dataset while preserving the class proportions. The dataset consisted of a total of 895 rows with a total of 1626 multi-label compositions across the samples.

The data composition shown in Figure 4 represents the distribution before stratified sampling. The initial distribution of multi-label classes was done before the dataset was divided. The dataset

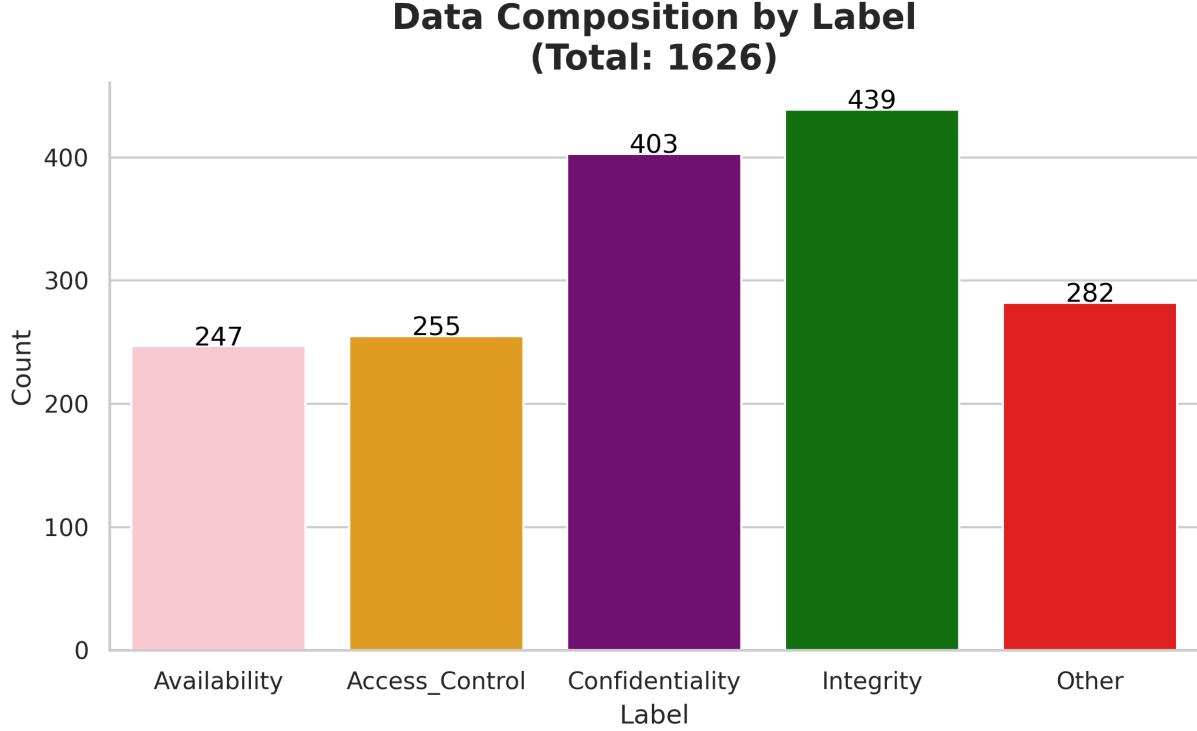


Figure 4: Data Composition by Label in the CWE Dataset (Total: 1626)

was first divided into a training set 80% and a combined validation and test set 20% while maintaining the proportional distribution of the multi-label classes. The validation and test sets were then further split into 15% and 5%, respectively, ensuring that the distribution remained consistent across all subsets. The stratified sampling method made sure that the training, validation, and test sets all had the same proportional class distribution. The dataset comprises five multi-label classes, each representing distinct cybersecurity consequences. The distribution of these labels is as follows: *Availability* (247 instances), *Access Control* (255 instances), *Confidentiality* (403 instances), *Integrity* (439 instances), and *Other* (282 instances). The total number of labeled instances in the dataset is **1,626**, as illustrated in the figure.

7 Analysis and Results

7.1 Label-Wise Performance

To provide a deeper insight into the model’s performance, evaluation metrics were calculated for each individual label. Table 1 reports these metrics, demonstrating that the model achieved the highest F1-scores for Confidentiality and others.

From the table, it is evident that the **Confidentiality** and **Other** labels exhibit the highest F1-scores, indicating superior model performance in predicting these categories. The **Access Control** label also performed commendably, with an F1-score of 0.9060. However, the **Integrity** label

Table 1: Label-Wise Performance Metrics of BERT Model

Label	Accuracy	Precision	Recall	F1-Score
Availability	0.9330	0.8627	0.8980	0.8800
Access Control	0.9385	0.8833	0.9298	0.9060
Confidentiality	0.9609	0.9538	0.9394	0.9466
Integrity	0.9050	0.8125	0.9123	0.8595
Other	0.9665	0.9625	0.9625	0.9625

showed a relatively lower F1-score, suggesting room for improvement in this area. The observed improvements are primarily attributed to the enhanced contextual understanding of BERT, which is crucial for interpreting cybersecurity texts that contain domain-specific terminology and complex sentence structures.

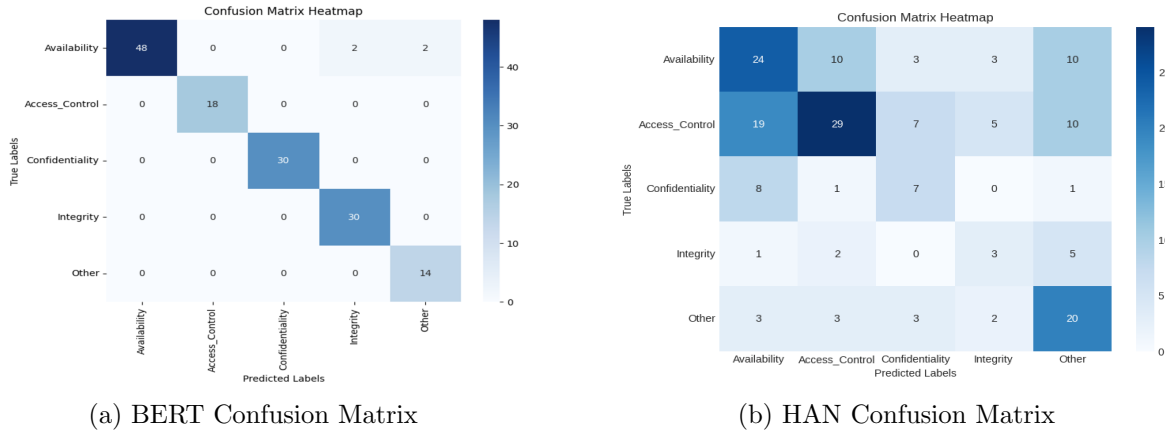


Figure 5: Confusion Matrices of Models on Five Labels

7.2 Confusion Matrix and Comparative Analysis

As shown in Figure 5, both the BERT model and the HAN was evaluated using confusion matrices. We analyzed the confusion matrix to gain further insight into the model’s classification performance. BERT model 5a demonstrates exceptional accuracy, correctly classifying 48 instances of availability with only four misclassifications, while access control, confidentiality, integrity, and others were classified with perfect accuracy, showing zero misclassifications.

In contrast, the HAN model 5b struggles with high misclassification rates, which is evident from the figure. Correctly identified 24 instances of availability but misclassified 26 others, primarily confusing them with Access Control 10 and Other 10. The Access Control category also performed poorly, with 29 correct predictions but 41 misclassifications, often mistaken for Availability 19 times. The figure further highlights HAN’s difficulty in classifying Confidentiality and Integrity, with only 7 and 3 correct classifications, respectively, and frequent misclassifications into other categories. Even the Other category, which had 20 correct classifications, still showed significant confusion across multiple labels.

This suggests that BERT effectively distinguishes between categories, handling the classification

task with remarkable precision, as illustrated in the figure. Overall, the figure confirms that BERT significantly outperforms HAN, making far fewer errors and classifying most categories correctly.

8 Performance Comparison

In this section, we compare the performance of our BERT-based model with the results obtained in the original paper [10], which used LSTM and CNN architectures for multi-label classification of cyberattack consequences. The comparison focuses on key metrics, including accuracy, precision, recall, and F1-score, for the five predicted labels: Availability, Access Control, Confidentiality, Integrity, and Other.

8.1 Overall Performance Comparison

The original paper [10] explored multiple architectures, including LSTM with multiple outputs, LSTM with a single output, CNN, CNN-LSTM, and CNN-BiLSTM. Table 2 shows the overall performance of the best model in the original paper (CNN-LSTM with random search) compared to our BERT-based model. Our BERT-based model outperformed the best model from the original paper in all metrics, particularly the F1-score.

Table 2: Overall Performance Comparison

Metric	CNN-LSTM (Original [10])	BERT (Ours)
Accuracy	0.4357	0.9722
Precision	0.64	0.9864
Recall	0.68	0.9731
F1-Score	0.72	0.9797

As shown in Table 2, our model achieves a significantly higher accuracy of 0.9722 compared to the 0.4357 obtained by the CNN-LSTM architecture. Similarly, our BERT model significantly improves precision, recall, and F1 score, demonstrating its superior ability to capture the relationships between labels.

8.2 Label-Wise Performance Comparison

Our models were evaluated on five cybersecurity consequence labels: availability, access control, confidentiality, integrity, and other—using precision, precision, recall, and F1 score as key evaluation metrics. Table 3 presents a comprehensive comparison of three models: the original CNN-LSTM-RS model and our fine-tuned BERT and HAN models for predicting the consequences of cyberattacks.

The BERT-based model demonstrates significant improvements over CNN-LSTM-RS, achieving an overall accuracy of 0.972, which is a substantial increase from the 0.436 reported in the original study. Its strongest performance gains are observed in Confidentiality and Other, where it attains F1-scores of 0.95 and 0.96, respectively. BERT’s transformer-based architecture enables

Table 3: Label-Wise Performance Comparison

Label	Metric	Original Paper (CNN-LSTM-RS) [10]	BERT (Ours)	HAN (Ours)
Availability	Accuracy	0.681	0.933	0.773
	Precision	0.53	0.86	0.61
	Recall	0.36	0.90	0.61
	F1-Score	0.42	0.88	0.61
Access Control	Accuracy	0.765	0.939	0.778
	Precision	0.56	0.88	0.64
	Recall	0.49	0.93	0.67
	F1-Score	0.52	0.91	0.65
Confidentiality	Accuracy	0.703	0.961	0.758
	Precision	0.73	0.95	0.79
	Recall	0.56	0.94	0.87
	F1-Score	0.63	0.95	0.83
Integrity	Accuracy	0.720	0.905	0.758
	Precision	0.78	0.81	0.77
	Recall	0.64	0.91	0.83
	F1-Score	0.70	0.86	0.80
Other	Accuracy	0.787	0.967	0.758
	Precision	0.64	0.96	0.59
	Recall	0.83	0.96	0.56
	F1-Score	0.72	0.96	0.57
Overall	Accuracy	0.436	0.972	0.444
	Precision (Micro)	0.64	0.99	0.71
	Recall (Micro)	0.68	0.97	0.75
	F1-Score (Micro)	0.72	0.98	0.73
	F1-Score (Macro)	-	0.98	0.69

it to effectively capture intricate contextual relationships in cybersecurity text, which enhances classification accuracy across all labels. By leveraging pre-trained embeddings and deep contextual representations, BERT effectively models the semantic dependencies between cyber vulnerabilities and their consequences, surpassing the capabilities of traditional deep learning models such as CNN and LSTM.

The HAN model, while not as dominant as BERT in overall classification, exhibits notable strengths in specific labels. With an overall accuracy of 0.444, HAN does not outperform BERT, but it exceeds CNN-LSTM-RS in the Confidentiality and Integrity categories. This suggests that HAN’s hierarchical attention mechanisms are particularly effective for cybersecurity text analysis, where the data exhibits structural and contextual dependencies. By processing key textual components and emphasizing essential phrases and dependencies, the HAN model effectively captures hierarchical relationships and salient features more effectively than sequence-based models like LSTM.

The confidentiality category score of 0.95 is a significant leap over the 0.63 achieved by CNN-LSTM-RS, further emphasizing the effectiveness of transformer models in extracting meaningful cybersecurity features from text descriptions. These results highlight the superiority of BERT in predictive accuracy when compared to conventional deep learning architectures that rely on sequential modeling.

Despite HAN’s moderate overall performance across all five labels, it delivers noteworthy improvement in two key categories: confidentiality and integrity, where its F1-scores surpass those of CNN-LSTM-RS. This suggests that HAN’s hierarchical structure and attention-based framework

are particularly well-suited for specific classification tasks, especially when structured text representations are crucial. Although BERT remains the most effective model for classifying general cybersecurity consequences, the targeted strengths of HAN suggest its potential for hybrid approaches, where its hierarchical capabilities could be leveraged in conjunction with state-of-the-art transformer models to enhance cybersecurity text classification.

9 Conclusion and Future Work

The paper presents the BERT-based Model and HAN model for predicting cyberattacks with five labels (i.e., Availability, Access Control, Confidentiality, Integrity, and Other) and two labels (i.e., Access Control and Integrity), respectively. In five labels, the BERT model achieved an accuracy of 0.972, precision of 0.99, recall of 0.97, and F1 score of 0.98. The HAN model achieved an 0.44 in accuracy, 0.71 in precision, 0.75 in recall and 0.73 in F1 score. The BERT-based model for predicting cyberattack consequences demonstrates significant improvements over previous methods CNN-LSTM-RS [10], particularly in terms of precision, recall, F1 score and accuracy.

The comparison between the original paper and our approach demonstrates the significant improvement brought by the BERT-based model. The original paper employed a combination of CNN and LSTM architectures, which struggled to capture complex relationships in text data. In contrast, the BERT model’s pre-trained transformer architecture is better suited for handling long-range dependencies in text, resulting in improved generalization and enhanced multi-label classification performance. Furthermore, using the BERT tokenizer and embeddings also helped people understand the descriptions better by providing more context, which led to higher F1-scores and overall metrics. The superior handling of complex cyberattack descriptions by the BERT model resulted in more accurate predictions of attack consequences across all five labels.

Despite the strong performance, the model has certain limitations. One notable challenge is data imbalance, where the *Integrity* label exhibited relatively lower performance, likely due to an uneven distribution of labels in the dataset. Techniques such as data augmentation) to balance the dataset and enhance performance for minority labels. Another limitation is related to long sequence handling, as the BERT model is constrained to processing sequences of up to 256 tokens. This poses a challenge for cybersecurity descriptions that exceed this length. Future research directions include exploring other transformer-based models, such as Alberta, TinyBert, and Roberta, and employing transfer learning techniques to enhance performance on smaller, imbalanced datasets.

Acknowledgment

This work is partially supported by a grant from the National Science Foundation (Award No. 2319802).

References

- [1] Olakunle Abayomi Ajala. Leveraging ai/ml for anomaly detection, threat prediction, and automated response. 2024.
- [2] Muhammad Shoaib Akhtar and Tao Feng. Malware analysis and detection using machine learning algorithms. *Symmetry*, 14(11):2304, 2022.
- [3] Amazon Web Services. How AWS Protects Customers from DDoS Events. *Online*, 2023. Available: <https://aws.amazon.com/blogs/security/how-aws-protects-customers-from-ddos-events/>.
- [4] Ahmed Bahaa, Aya El-Rahman Kamal, Hanan Fahmy, and Amr S Ghoneim. Db-cbil: A distilbert-based transformer hybrid model using cnn and bilstm for software vulnerability detection. *IEEE Access*, 2024.
- [5] Deepali Bajaj, Urmil Bharti, Isha Gupta, Priya Gupta, and Asha Yadav. Gtmi-cro—microservice identification approach based on deep nlp transformer model for greenfield developments. *International Journal of Information Technology*, 16(5):2751–2761, 2024.
- [6] Mhamad Bakro, Rakesh Ranjan Kumar, Amerah Alabrah, Zubair Ashraf, Md Nadeem Ahmed, Mohammad Shameem, and Ahmed Abdelsalam. An improved design for a cloud intrusion detection system using hybrid features selection approach with ml classifier. *IEEE Access*, 11:64228–64247, 2023.
- [7] Chaoqi Chen, Yushuang Wu, Qiyuan Dai, Hong-Yu Zhou, Mutian Xu, Sibe Yang, Xiaoguang Han, and Yizhou Yu. A survey on graph neural networks and graph transformers in computer vision: A task-oriented perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [8] Lu Cheng, Ruocheng Guo, Yasin N. Silva, Deborah Hall, and Huan Liu. Modeling temporal patterns of cyberbullying detection with hierarchical attention networks. *ACM/IMS Trans. Data Sci.*, 2(2), April 2021.
- [9] Rahul Dass et al. Applying hidden markov models for cyber attack impact prediction. In *Proceedings of the International Conference on Security and Privacy in Communication Networks*, pages 45–62. Springer, 2023.
- [10] Prerit Datta, Akbar Siامي Namin, and Keith S Jones. Can we predict consequences of cyber attacks? In *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1047–1054. IEEE, 2022.
- [11] Suman Datta et al. Predicting cyber attack consequences using machine learning techniques. *ACM Transactions on Cybersecurity*, 3(1):1–25, 2023.

- [12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.
- [13] Shiv Ram Dubey and Satish Kumar Singh. Transformer-based generative adversarial networks in computer vision: A comprehensive survey. *IEEE Transactions on Artificial Intelligence*, 2024.
- [14] Abdelkader El Mahdaouy, Salima Lamsiyah, Meryem Janati Idrissi, Hamza Alami, Zakaria Yartaoui, and Ismail Berrada. Domurls_bert: Pre-trained bert-based model for malicious domains and urls detection and classification. *arXiv preprint arXiv:2409.09143*, 2024.
- [15] Hugging Face. Bert-base, uncased, 2019. Accessed: 2024-11-20.
- [16] Housseem Gasmi, Jannik Laval, and Abdelaziz Bouras. Information extraction of cybersecurity concepts: An lstm approach. *Applied Sciences*, 9(19):3945, 2019.
- [17] Housseem Gasmi, Jannik Laval, and Abdelaziz Bouras. Lstm recurrent neural networks for cybersecurity named entity recognition. *arXiv preprint arXiv:2409.10521*, 2024.
- [18] Google Cloud. How It Works: The Novel HTTP/2 Rapid Reset DDoS Attack. *Online*, 2023. Available: <https://cloud.google.com/blog/products/identity-security/how-it-works-the-novel-http2-rapid-reset-ddos-attack>.
- [19] Saroj Gopali, Zulfiqar Ali Khan, Bipin Chhetri, Bimal Karki, and Akbar Siami Namin. Vulnerability detection in smart contracts using deep learning. In *2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 1249–1255. IEEE, 2022.
- [20] Saad Hikmat Haji and Siddeeq Y Ameen. Attack and anomaly detection in iot networks using machine learning techniques: A review. *Asian J. Res. Comput. Sci*, 9(2):30–46, 2021.
- [21] Lei Han et al. Constructing cybersecurity knowledge graphs for threat intelligence and prediction. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):1025–1038, 2023.
- [22] HIPAA Journal. Change Healthcare Responding to Cyberattack. *Online*, 2024. Available: <https://www.hipaajournal.com/change-healthcare-responding-to-cyberattack/>.
- [23] Homa Hosseinmardi, Sabrina Arredondo Mattson, Rahat Ibn Rafiq, Richard Han, Qin Lv, and Shivakant Mishra. Detection of cyberbullying incidents on the instagram social network. *arXiv preprint arXiv:1503.03909*, 2015.

- [24] Ye Hui, Lin Du, ShuYuan Lin, YiQian Qu, and Dong Cao. Extraction and classification of tcm medical records based on bert and bi-lstm with attention mechanism. In *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1626–1631. IEEE, 2020.
- [25] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics, 2014.
- [26] Geeta Kocher and Gulshan Kumar. Machine learning and deep learning methods for intrusion detection systems: recent developments and challenges. *Soft Computing*, 25(15):9731–9763, 2021.
- [27] Shaswata Mitra, Stephen A Torri, and Sudip Mittal. Survey of malware analysis through control flow graph using machine learning. In *2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 1554–1561. IEEE, 2023.
- [28] MITRE. CWE - Home. *Online*, 2025. Available: <https://cwe.mitre.org/about/index.html>.
- [29] Salman Muneer, Umer Farooq, Atifa Athar, Muhammad Ahsan Raza, Taher M Ghazal, and Shadman Sakib. A critical review of artificial intelligence based approaches in intrusion detection: A comprehensive analysis. *Journal of Engineering*, 2024(1):3909173, 2024.
- [30] Loc Gia Nguyen and Kohei Watabe. Flow-based network intrusion detection based on bert masked language model. CoNEXT-SW '22, page 7–8, New York, NY, USA, 2022. Association for Computing Machinery.
- [31] Abir Rahali and Moulay A Akhloufi. End-to-end transformer-based models in textual-based nlp. *Ai*, 4(1):54–110, 2023.
- [32] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.
- [33] Scikit-learn Developers. Scikit-learn: Machine Learning in Python. *Online*, 2025. Available: https://scikit-learn.org/stable/modules/model_evaluation.html.
- [34] Frank Swiderski and Window Snyder. *Threat modeling*. Microsoft Press, 2004.
- [35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

- [36] Yuanyuan Wei, Julian Jang-Jaccard, Wen Xu, Fariza Sabrina, Seyit Camtepe, and Mikael Boulic. Lstm-autoencoder-based anomaly detection for indoor air quality time-series data. *IEEE Sensors Journal*, 23(4):3787–3800, 2023.
- [37] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. 2024.
- [38] Chenjian Xu, Weirui Sun, and Mengxue Li. Dtt: A dual-domain transformer model for network intrusion detection. *EAI Endorsed Transactions on Scalable Information Systems*, 11(6), May 2024.
- [39] Surendra Yadav, Hina Hashmi, Daxa Vekariya, et al. Mitigation of attacks via improved network security in iot network environment using rnn. *Measurement: Sensors*, 32:101046, 2024.
- [40] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489, 2016.