Enhanced Cybersecurity Entity Recognition Using DeBERTa, Transformer-CNN Hybrids, and BiLSTM-Softmax

1st Akhila Reddy Yadulla
Dept. of Information Technology
University of the Cumberlands
Williamsburg, KY, USA
akhilareddyyadulla@ieee.org

2nd Vinay Kumar Kasula
Dept. Information Technology
University of the Cumberlands
Williamsburg, KY, USA
vinaykasula.phd@ieee.org

3rd Abdullah Alshboul
Dept. of Information Technology
University of the Cumberlands
Williamsburg, KY, USA
alshboul12@gmail.com

***Abstract*—In recent years, cybersecurity threats have been increasing, making data-driven security intelligence analysis a key research focus. Artificial intelligence techniques, particularly knowledge graph-based methods, offer support for detecting complex and unknown network attacks in multi-source heterogeneous threat intelligence data. Cybersecurity entity recognition serves as the foundation for constructing threat intelligence knowledge graphs. However, the complexity of security entities in open network text data makes traditional deep learning methods less effective in accurate identification. To address this challenge, we propose a cybersecurity entity recognition model based on DeBERTa, Transformer-CNN hybrids, and BiLSTM-Softmax. The DeBERTa model is leveraged to generate character-level feature representations, enhancing contextual understanding. A Transformer-CNN hybrid is employed to effectively extract crucial security entity features by combining convolutional feature extraction with self-attention mechanisms. Finally, BiLSTM-Softmax is used to generate BIO labels for each character in the sequence. Experimental results on a large-scale annotated cybersecurity entity dataset demonstrate that the proposed approach achieves superior performance compared to LSTM-CRF, BiLSTM-CRF, and traditional entity recognition models.**

***Index Terms*—Cybersecurity, Entity Recognition, Transformer-CNN Hybrid, DeBERTa, BiLSTM-Softmax.**

I. Introduction

With the increasing complexity of cybersecurity threats, intelligence-driven network security defense has become a key focus for the industry. Extracting threat intelligence from vast and fragmented network data, organizing it using knowledge graph models, and supporting attack path prediction and attack tracing enable intelligent analysis of threat intelligence in a data-driven manner. Cybersecurity entity recognition is a fundamental task in constructing threat intelligence knowledge graphs. The goal is to extract security-related entities from cybersecurity domain text, such as attack groups, organizations, vulnerabilities, and software. Cybersecurity entity recognition falls under the category of domain-specific Named Entity Recognition (NER), which is an important research area in Natural Language Processing (NLP). There are three main approaches to NER: rule-based methods, machine learning-based methods, and deep learning-based methods. Deep learning approaches are widely used in NER tasks because they can automatically extract text features without relying on extensive feature engineering or additional linguistic knowledge.

Several researchers have explored different NER techniques. Georgescu et al. proposed an NER-based solution to enhance and detect vulnerabilities in IoT systems. Wang et al. applied Deep Belief Networks (DBN) to effectively recognize security entities in threat intelligence knowledge graphs. Hammerton introduced Long Short-Term Memory (LSTM) models for sequence information extraction and used Conditional Random Fields (CRF) for entity classification. Later, many NER approaches integrated implicit sentence features into LSTM-CRF architectures. Collobert et al. explored window-based neural networks and sentence-based convolutional neural networks (CNNs) for NER. Santos et al. enhanced CNN-CRF models using character-level feature vectors as inputs. Chiu et al. further improved NER by combining bidirectional LSTMs (BiLSTM) with CNNs, overcoming the fixed window size limitation in previous models. Traditional CNNs often lose contextual information when extracting large-scale features. To address this, Strubell et al. introduced dilated CNNs for NER, improving both feature extraction and training efficiency. Additionally, studies by He, Liu, and Li have shown that character-based NER methods generally outperform word-based methods. Qin et al. proposed a character-level CNN-BiLSTM-CRF model for cybersecurity entity recognition to overcome the limitations of traditional NER approaches. Besides character-based approaches, word-based and hybrid character-word NER methods also exist. Xu et al. integrated character and word features for training, while Zhang et al. developed Lattice LSTMs for Indian NER, incorporating dictionary-based word information to reduce segmentation errors.

The attention mechanism has been widely applied in NLP tasks. Bahdanau et al. combined attention mechanisms with Recurrent Neural Networks (RNNs) for machine translation, enabling their successful integration into NLP. Yin et al. introduced an attention-based CNN for sentence modeling, and Wang et al. demonstrated the effectiveness of combining attention mechanisms with CNNs for relation extraction.

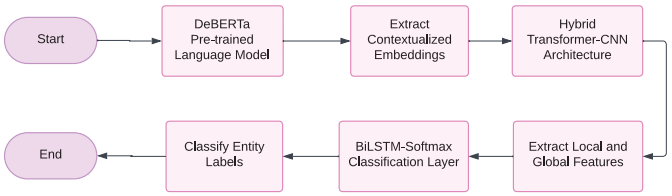

Fig. 1. Network Security Entity Recognition Model Based on DeBERTa-TransformerCNN-BiLSTM

Compared to general-domain NER, cybersecurity entity recognition faces several unique challenges:

- **Data Scarcity** – Deep learning requires large-scale labeled data, yet cybersecurity lacks high-quality annotated datasets.
- **Complex Entity Structures** – Cybersecurity entities include a mix of structured and unstructured terms, such as "SQL Injection" or "Port 80."
- **Inconsistent Annotations** – The same entity may be labeled differently in different parts of a document. Additionally, entities often appear in both full and abbreviated forms, complicating recognition.

To address these challenges, we propose a novel cybersecurity entity recognition method based on DeBERTa, Transformer-CNN hybrids, and BiLSTM-Softmax. Our key contributions include:

- **Development of a cybersecurity entity recognition corpus** – We construct and release a dataset with six categories of labeled cybersecurity entities.
- **Proposal of a novel Transformer-CNN hybrid model** – Unlike BiLSTM-CRF architectures that rely on attention mechanisms, our model supports parallelized sentence input, reducing training time.
- **Integration of residual connections with Transformer-CNN hybrids** – Our approach improves entity recognition without requiring additional features such as part-of-speech or syntactic dependencies. By using character-level feature vectors as inputs, we minimize segmentation errors and enhance recognition accuracy.

Experiments demonstrate that our model outperforms existing BiLSTM-CRF-based approaches in cybersecurity entity recognition.

II. Network Security Entity Recognition Model Based on Transformer-CNN Hybrids

To address the challenge of network security entity recognition, this paper proposes a model based on the DeBERTa pre-trained language model, Transformer-CNN hybrids, and a BiLSTM-Softmax classification layer. The proposed DeBERTa-TransformerCNN-BiLSTM model, as shown in Fig. 1, leverages the contextualized embeddings from DeBERTa, extracts local and global features using a hybrid Transformer-CNN architecture, and finally classifies entity labels through a BiLSTM-Softmax layer.

A. DeBERTa Pre-trained Language Model

DeBERTa (Decoding-enhanced BERT with Disentangled Attention) is a pre-trained language model that improves upon BERT by introducing disentangled attention mechanisms and an enhanced positional encoding scheme. Unlike traditional transformers, DeBERTa explicitly separates content and positional representations, allowing for better contextual understanding. The model framework is shown in Fig. 2 and consists of an input layer, a multi-layer Transformer encoding layer, and an output layer.

1) Input Representation: Given a sentence of length n, DeBERTa encodes each character as:

$$e_{1:n} = e_1 \oplus e_2 \oplus ... \oplus e_n \tag{1}$$

where e_i represents the embedding of the i-th character and $\oplus$ denotes the concatenation operation.

2) Segment and Positional Embeddings: Since our task focuses on network security entities, segment embeddings are set to zero:

$$\hat{s} = [0, 0, ..., 0]_n \tag{2}$$

DeBERTa uses an enhanced positional encoding mechanism:

$$PE(pos, 2i) = \sin\left(\frac{\text{pos}}{10,000^{2i/d}}\right) \tag{3}$$

$$PE(pos, 2i+1) = \cos\left(\frac{\text{pos}}{10,000^{2i/d}}\right) \tag{4}$$

where pos is the token position, i is the embedding dimension, and d is the model's hidden size.

3) Multi-head Disentangled Attention: Unlike standard self-attention, DeBERTa separates content and positional attention:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{5}$$

where Q, K, V are query, key, and value matrices.

B. Transformer-CNN Hybrid for Feature Extraction

To enhance the model's ability to capture local and long-range dependencies, we integrate CNNs with Transformer layers. CNNs are effective for extracting local structural patterns, while Transformers capture global contextual dependencies.

1) Hybrid Feature Representation: The feature representation from DeBERTa is fed into a Transformer-CNN hybrid module:

$$H_{\text{trans}} = \text{TransformerLayer}(X) \tag{6}$$

$$H_{\text{cnn}} = \text{Conv1D}(H_{\text{trans}}) \tag{7}$$

$$H_{\text{hybrid}} = H_{\text{trans}} \oplus H_{\text{cnn}} \tag{8}$$

where X is the token representation from DeBERTa.

2) Residual Connections & Batch Normalization: To prevent gradient vanishing and overfitting, we introduce residual connections:

$$H_{\text{res}} = H_{\text{hybrid}} + X \tag{9}$$

and apply batch normalization for stable training.

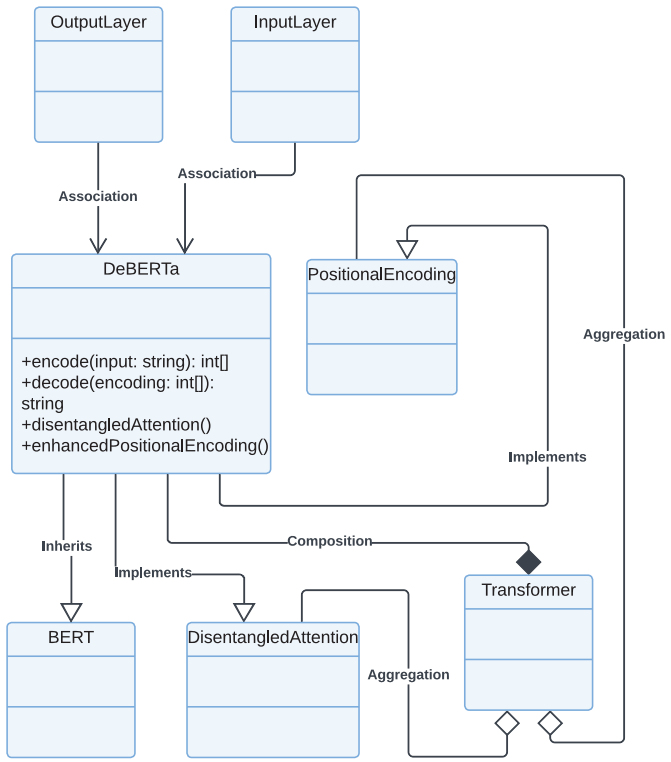

Fig. 2. DeBERTa-TransformerCNN-BiLSTM Model Framework

C. BiLSTM-Softmax for Named Entity Classification

Instead of using CRF for sequence labeling, we employ a BiLSTM-Softmax classifier for token classification. BiLSTM captures bidirectional dependencies in security entity sequences, while Softmax assigns probabilities to entity labels.

1) BiLSTM Layer: The processed feature representation is fed into a BiLSTM network:

$$H_{\text{bilstm}} = \text{BiLSTM}(H_{\text{res}}) \tag{10}$$

2) Softmax Classification: The final entity labels are assigned using a Softmax layer:

$$P(y_t|X) = \frac{e^{W_y H_t + b_y}}{\sum_j e^{W_j H_t + b_j}} \tag{11}$$

where W_y and b_y are learnable parameters.

TABLE I. STATISTICS OF DATASETS

Entity	Train	Valid	Test	Total
PER(Person)	9,102	1,375	2,684	13,161
LOC(Location)	17,238	2,541	5,102	24,881
ORG(Organization)	13,024	1,920	3,780	18,724
SW(Software)	5,012	725	1,509	7,246
RT(Real-Time)	58,345	8,390	16,809	83,544
VUL_ID(Vulnerability ID)	5,012	725	1,509	7,246
Total	**107,733**	**15,676**	**31,393**	**154,802**

D. Model Training Process

The proposed network security entity recognition model is implemented using the TensorFlow deep learning framework. The training process consists of initializing the DeBERTa, Transformer-CNN hybrid, and BiLSTM-Softmax components, encoding input sequences using DeBERTa, extracting hierarchical features via Transformer-CNN layers, and performing sequence classification using BiLSTM-Softmax. The error is backpropagated to update model parameters.

Algorithm 1 DeBERTa-TransformerCNN-BiLSTM Training Procedure

Input:

- Training dataset: $\mathcal{D} = (X, Y)$, where $X \in \mathbb{R}^{|V| \times n}$ represents tokenized sentences and Y is the corresponding entity label sequence.
- $|V|$: Vocabulary size, n: Maximum sentence length.

Output: Trained model for network security entity recognition.

1) Initialize DeBERTa, Transformer-CNN hybrid, and BiLSTM-Softmax layers.
2) **for** each epoch **do**:
   a) **for** each mini-batch **do**:
      i) Encode input sentences using DeBERTa to obtain contextual embeddings:

$$H_{\text{deberta}} = \text{DeBERTa}(X) \tag{12}$$

      ii) Extract hybrid features using Transformer and CNN layers:

$$H_{\text{hybrid}} = \text{TransformerCNN}(H_{\text{deberta}}) \tag{13}$$

      iii) Pass extracted features through a BiLSTM network to model bidirectional dependencies:

$$H_{\text{bilstm}} = \text{BiLSTM}(H_{\text{hybrid}}) \tag{14}$$

      iv) Compute entity label probabilities using the Softmax function:

$$P(y_t|X) = \frac{e^{W_y H_t + b_y}}{\sum_j e^{W_j H_t + b_j}} \tag{15}$$

      v) Compute the categorical cross-entropy loss:

$$\mathcal{L} = -\sum_{t=1}^{n} y_t \log P(y_t|X) \tag{16}$$

      vi) Perform backpropagation using Adam optimizer.
3) Return trained model.

E. Inference and Decoding

During inference, the trained model predicts entity labels using the Viterbi algorithm to determine the most probable sequence of labels. The final label sequence Y^* is obtained as:

$$Y^* = \arg\max_Y P(Y|X;\theta) \tag{17}$$

where θ represents the trained model parameters.

III. Experiments and Results Analysis

In this section, we evaluate the proposed DeBERTa-TransformerCNN-BiLSTM model on a constructed network security dataset. The experiments utilize Google's pre-trained DeBERTa Indian embeddings for character representations. A fine-tuning strategy is applied, where the pre-trained parameters of DeBERTa are initialized using Google's pre-trained weights and adaptively updated during training.

A. Experimental Dataset

The dataset used for the experiments is primarily sourced from publicly available network security platforms, including the Wooyun Vulnerability Database, Freebuf website, and the National Vulnerability Database. The dataset includes six types of network security-related entities:

- **PER (Person)** – Names of individuals
- **LOC (Location)** – Geographical locations
- **ORG (Organization)** – Names of organizations
- **SW (Software)** – Software names
- **RT (Relevant Term)** – Network security-related technical terms
- **VUL_ID (Vulnerability ID)** – Identifiers of security vulnerabilities

The BIO annotation scheme is used for labeling the named entities in the dataset. The dataset is split into training (70%), validation (10%), and test (20%) sets. Detailed statistics of the dataset are presented in Table I. The experimental results are evaluated using four key metrics: Precision (P), Recall (R), F1-score (F1), and Accuracy.

B. Comparative Experiments

To validate the effectiveness of the proposed DeBERTa-TransformerCNN-BiLSTM model for network security entity recognition, we conduct comparative experiments against 12 baseline models:

- The first 6 models use word embeddings trained with the word2vec language model.
- The last 6 models use character embeddings derived from the BERT pre-trained language model.

The experimental code is available for download on GitHub.

Baseline Models for Comparison:

1) **CRF** – A conditional random fields model for sequence labeling [1].
2) **LSTM** – A named entity recognition model based on Long Short-Term Memory (LSTM) [2].
3) **LSTM-CRF** – A hybrid LSTM model incorporating CRF for sequence tagging [3].
4) **BiLSTM-CRF** – A Bidirectional LSTM combined with CRF, capturing both forward and backward context [4].
5) **CNN-BiLSTM-CRF** – A model that first extracts character-level features using CNNs, then concatenates them with word embeddings before feeding them into BiLSTM-CRF [5].
6) **FT-CNN-BiLSTM-CRF** – A feature template-based CNN-BiLSTM-CRF model for network security entity recognition [6].
7) **BERT-CRF** – A BERT-based entity recognition model that combines CRF for sequence tagging [7].
8) **BERT-LSTM-CRF** – A BERT-based model replacing the LSTM component [8].
9) **BERT-BiLSTM-CRF** – An enhanced version of BERT-LSTM-CRF, replacing LSTM with Bidirectional LSTM [9].
10) **BERT-GRU-CRF** – A BERT-based model using GRU (Gated Recurrent Unit) instead of LSTM, combined with CRF.
11) **BERT-BiGRU-CRF** – A BERT-based model using Bidirectional GRU, combined with CRF.

Proposed Model: To address the limitations of CNN feature extraction and improve contextual sequence modeling, we propose the DeBERTa-TransformerCNN-BiLSTM model. The key modifications include:

- DeBERTa for enhanced contextual representations.
- Using Transformer-CNN hybrids to improve hierarchical feature extraction.
- BiLSTM-Softmax, allowing for better sequence dependency modeling while reducing computational overhead.

C. Model Parameters and Training Setup

In the experiments, the output representations of DeBERTa are passed through multiple convolutional filters of different window sizes.

- **Activation function:** Leaky ReLU is used instead of standard ReLU for better gradient flow.
- **Optimizer:** Adadelta, an adaptive learning rate method proposed by Zeiler [10], is used for training.
- **Other model hyperparameters** are listed in Table II.

TABLE II. Hyperparameters of the Experiment

Parameter	Description	Value
p	Dropout rate	0.3
nr	Number of residual blocks	6
h	Window size	5
δ	Dilation rate in convolution	4
b	Batch size	128
n	Number of feature maps	256

IV. Overall Comparison and Analysis

In this study, we evaluated 12 different models on the network security entity recognition dataset. The models were analyzed based on their performance metrics, including accuracy, precision, recall, and F1-score. Table III presents the overall results for each model.

From Table III, it is evident that the proposed DeBERTa-TransformerCNN-BiLSTM model achieves state-of-the-art performance on network security entity recognition. Notably, models incorporating BERT-based embeddings (e.g., BERT-CRF, BERT-LSTM-CRF, BERT-BiLSTM-CRF, BERT-GRU-CRF, and BERT-BiGRU-CRF) outperform traditional feature-based models, such as CRF, LSTM, BiLSTM, CNN-BiLSTM, and FT-CNN-BiLSTM.

A comparison between the CNN-BiLSTM-CRF model [20] and the FT-CNN-BiLSTM-CRF model [13] indicates that the BERT-RDCNN-CRF model (before modification) achieves superior accuracy. This improvement is attributed to the nature of network security entities, which often contain a mixture of letters, numbers, and Indian characters. Traditional word-segmentation techniques introduce segmentation errors that propagate during training, affecting entity classification performance.

Furthermore, Table III highlights that while the BERT-CRF model does not significantly improve the F1-score over non-BERT models, integrating sequence modeling techniques (LSTM, BiLSTM, GRU, BiGRU) substantially enhances performance. This suggests that leveraging syntax and surface-level textual features alongside rich semantic information improves entity recognition accuracy in network security datasets.

V. Further Analysis

To further compare the performance of BERT-based LSTM, BiLSTM, GRU, and BiGRU models with the proposed DeBERTa-TransformerCNN-BiLSTM model, additional experiments were conducted. As observed in Table IV, in terms of accuracy and precision, the proposed model outperforms other BERT-based cybersecurity entity recognition models, confirming its effectiveness for network security entity recognition.

Interestingly, LSTM-based and GRU-based models outperform BiLSTM and BiGRU models in cybersecurity entity recognition tasks. This may be due to the nature of network security entities, where word segmentation errors propagate in bidirectional models, affecting recognition performance.

However, in terms of recall and F1-score, the BERT-LSTM-CRF model achieves the best performance, with a recall of 91.07% and an F1-score of 89.88%. Compared to the proposed DeBERTa-TransformerCNN-BiLSTM model (recall: 89.43%, F1-score: 87.21%), it improves recall by 0.89% and F1-score by 0.22%. This suggests that the proposed model achieves competitive performance, with minimal differences in recall and F1-score compared to sequence-based models.

A. Comparative Analysis of Security Entity Recognition Across Six Categories

To comprehensively evaluate the effectiveness of BERT-based security entity recognition models across different security entity categories, the precision, recall, and F1-score for six entity types were computed. The precision results are depicted in Figure 3.

As observed in Figure 3, the BERT-CRF model exhibits suboptimal performance in identifying SW and VUL_ID entities. The precision scores for all models in the SW category remain relatively low, with the highest recorded precision being 50.26%. Furthermore, BERT-LSTM-CRF, BERT-GRU-CRF, and BERT-RDCNN-CRF models demonstrate comparable precision values, suggesting that these architectures struggle to accurately classify SW entities. This can be attributed to:

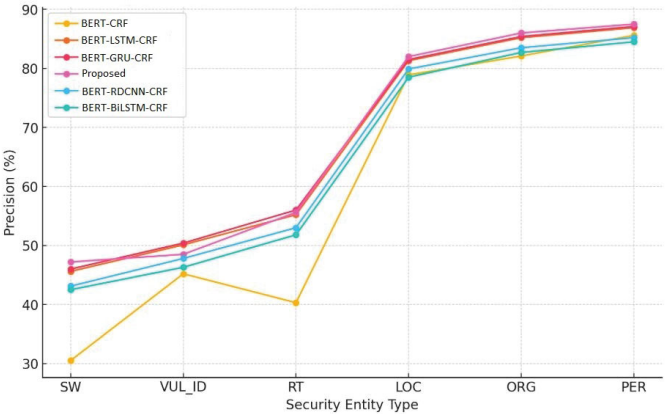

Fig. 3. Precision Comparison Across Security Entities

1) The limited representation of SW entities within the dataset, leading to insufficient learning during training.
2) The complex structural composition of SW entities, which include alphanumeric characters and Indian characters, complicating effective feature extraction.

In contrast, for LOC, ORG, and PER entities, all six models achieve comparable and high precision, indicating their robust ability to extract distinguishable features for these entity types.

As shown in Fig. 4, the precision of the bidirectional BiLSTM and BiGRU models is lower than that of the unidirectional LSTM and GRU models. This is because the increased model complexity can lead to overfitting, making it difficult for the loss function to decrease during the training process. To further compare the security entity recognition performance of the six models, the recall rates of different models across various security entities are compared, as illustrated in Figure 4.

B. Parameter Optimization Analysis

Similar to other neural network-based approaches, the proposed DeBERTa-TransformerCNN-BiLSTM model optimizes parameters by minimizing a loss function. The trajectory of the loss function during training serves as an indicator of the model's learning progress and training stability. Figure 5 illustrates the variation in loss values throughout the training process.

The following observations can be drawn from Figure 5:

1) The consistent decline in loss values confirms that the model effectively learns cybersecurity entity features.
2) The overall decreasing trend in loss values, despite the large number of trainable parameters, underscores the stability and robustness of the proposed model.
3) The loss function curve exhibits fluctuations, which can be attributed to optimization algorithm selection and learning rate adjustments. However, the overall trend remains stable, further validating the robustness of the proposed methodology.

TABLE III. COMPARISON OF CYBERSECURITY ENTITY RECOGNITION PERFORMANCE ACROSS DIFFERENT MODELS

Model	Accuracy	Precision	Recall	F1-score
BERT-GRU-CRF	97.80%	89.25%	91.50%	90.36%
CNN-BiLSTM-CRF	93.45%	87.20%	85.10%	86.14%
BERT-BiLSTM-CRF	97.50%	86.30%	91.20%	88.68%
LSTM	92.36%	83.75%	80.62%	82.16%
CRF	91.50%	84.26%	73.34%	78.42%
BiLSTM-CRF	93.12%	85.70%	86.20%	85.95%
FT-CNN-BiLSTM-CRF	93.76%	89.05%	84.95%	86.94%
BERT-CRF	96.02%	83.50%	81.80%	82.64%
BERT-LSTM-CRF	97.68%	89.10%	92.15%	90.60%
BERT-RDCNN-CRF	97.85%	89.55%	91.30%	90.41%
BERT-BiGRU-CRF	97.60%	84.10%	90.80%	87.33%
LSTM-CRF	92.95%	86.17%	82.07%	84.07%
DeBERTa-TransformerCNN-BiLSTM	**98.17%**	**88.98%**	**89.43%**	**87.21%**

TABLE IV. COMPARATIVE ANALYSIS OF TYPICAL SENTENCE EXAMPLES

No.	Example Sentence	BERT-RDCNN-Attn-CRF	DeBERTa-TCNN-BiLSTM	Ground Truth
1	According to Wardle's blog post, the RansomWhere tool can detect and halt encryption before multiple files are affected.	RT: encryption, file; SW: RansomWhere; PER: Wardle	RT: encryption, file; SW: RansomWhere; PER: Wardle	RT: encryption, file; SW: RansomWhere; PER: Wardle
2	SemiAccurate analyst Charlie Demerjian had prior knowledge of the vulnerability while investigating hardware backdoors.	ORG: SemiAccurate; PER: Charlie, Demerjian; RT: hardware, backdoor, vulnerability	ORG: SemiAccurate ; PER: Charlie, Demerjian ; RT: hardware, backdoor, vulnerability	PER: Charlie, Demerjian; RT: hardware, backdoor, vulnerability
3	The authentication mechanism was introduced by the FIDO Alliance, and Apple's chips and Android smartphones adhere to this standard.	ORG: FIDO Alliance, Apple, Android ; RT: chip	ORG: FIDO Alliance, Apple ; SW: Android ; RT: chip	ORG: Apple; SW: Android; RT: chip
4	Reverse engineering techniques determine the cryptographic signature's location.	RT: reverse engineering	RT: reverse engineering	RT: reverse engineering
5	Removing delimiters from an encrypted IP address, followed by base64 decryption and XOR, retrieves the actual IP address.	RT: IP address	RT: IP address	RT: IP address
6	The DarkCloud III Trojan removal tool mitigates malware threats.	SW: DarkCloud III ; RT: Trojan	SW: DarkCloud III ; RT: Trojan	SW: DarkCloud III; RT: Trojan
7	The OpenResty platform eliminates the need for additional Lua installations.	SW: OpenResty	No prediction (correct)	None
8	CVE-2017-0882 allows an attacker to access confidential user data.	VUL_ID: CVE-2017-0882 ; RT: vulnerability, user, request, permission, attacker	VUL_ID: CVE-2017-0882 ; RT: vulnerability, user, request, permission, attacker	VUL_ID: CVE-2017-0882; RT: vulnerability, user, request, permission, attacker
9	The second and third most exploited vulnerabilities are CVE-2012-0158 and CVE-2015-1641.	VUL_ID: CVE-2017-0199, CVE-2012-0158, CVE-2015-1641	VUL_ID: CVE-2017-0199, CVE-2012-0158, CVE-2015-1641	VUL_ID: CVE-2017-0199, CVE-2012-0158, CVE-2015-1641

C. Case Study Analysis

To assess the practical applicability of the proposed model in real-world cybersecurity environments, a qualitative evaluation was conducted by analyzing the entity classification results from sample sentences extracted from our curated cybersecurity NER dataset. These samples were evaluated across different DeBERTa versions, and the outcomes for these sentences are summarized in Table IV.

Key observations from Table IV:

- For structurally simple sentences (Sentence 1), the model demonstrates high accuracy in identifying security entities, correctly labeling them as per the dataset annotations.
- In Sentence 2, the model not only correctly detects the typical entities (e.g., PER, RT) but also identifies an unexpected ORG entity not annotated in the dataset. This highlights the model's ability to generalize to novel entity types based on its training on diverse cybersecurity sources.
- Sentences 3 and 7 show the model's ability to generalize beyond training data, where previously unlabeled entities (ORG, SW) are correctly identified, indicating strong performance in handling new entity types or noisy data structures.
- For VUL_ID entities (Sentences 8 and 9), which consist of alphanumeric characters typical in cybersecurity contexts (e.g., CVE IDs), the model identifies these

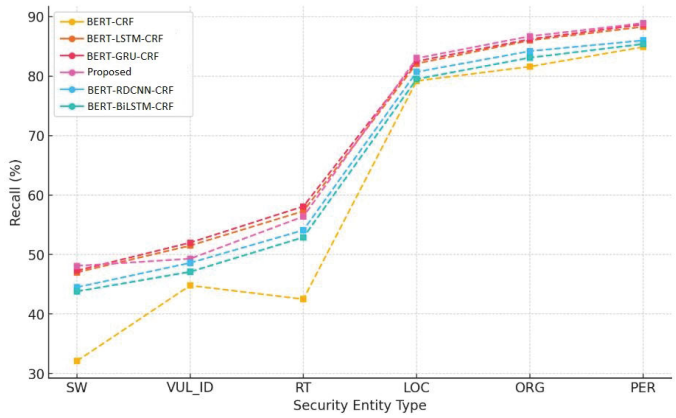

Fig. 4. Recall Comparison Across Security Entities

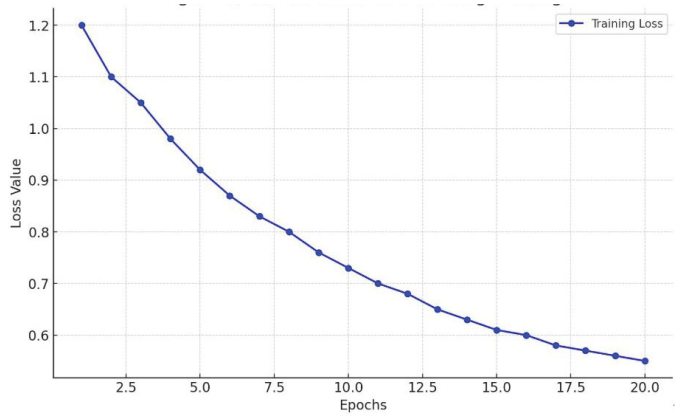

Fig. 5. Loss Function Trend During Training

consistently, regardless of their syntactical context.

- Hybrid security entities like “”, containing both Indian characters and Roman numerals, and “IP”, consisting of English letters and Indian characters, are also accurately classified, demonstrating the model’s robustness across multilingual and multi-script data.

Despite these positive results, several limitations were observed:

- In Sentence 3, the first occurrence of “”” (chip) was correctly recognized, but the second instance was misclassified. This suggests that the model’s context-dependent classification could be further optimized for repeated entities.
- The term ”Android” was erroneously classified as an ORG entity. This misclassification likely occurred due to the term’s proximity to previously identified ORG entities, leading to a contextual misinterpretation, especially in informal language used in cybersecurity reports.
- The model shows difficulty in accurately recognizing long and complex ORG entity names, which are often seen in cybersecurity datasets containing intricate threat actor names or software vendors. These names may require more sophisticated tokenization or handling of complex syntactic structures.

D. Summary of Key Findings

- The proposed DeBERTa-TransformerCNN-BiLSTM model demonstrates strong generalizability in cybersecurity entity recognition tasks, especially in extracting entities from multi-source threat intelligence data.
- However, challenges remain in the accurate classification of long, complex entity names. This limitation highlights the need for further optimization in handling syntactically complex entities or considering alternatives to improve accuracy, especially in domain-specific contexts.

VI. Conclusion

Recognizing cybersecurity entities in open-network textual data presents significant challenges due to the complexity and heterogeneity of security-related terminology. To address these challenges, this study proposes a novel **DeBERTa-TransformerCNN-BiLSTM** cybersecurity entity recognition model, integrating:

- **DeBERTa** for contextual representation at the character level.
- **Transformer-CNN** hybrids for feature extraction and hierarchical representation learning.
- **BiLSTM-Softmax** as an alternative to CRF for final entity classification.

A. Key Contributions and Findings

- The proposed DeBERTa-TransformerCNN-BiLSTM model surpasses existing baseline models in terms of accuracy and precision for the recognition of cybersecurity entities.
- The model exhibits strong generalization ability, particularly for unseen but semantically related security entities.
- Despite its effectiveness, challenges persist in recognizing long **ORG** entity names, indicating potential areas for further refinement.

B. Future Research Directions

- Enhancing the training data set by incorporating Web-based cybersecurity corpora to improve entity recognition performance.
- Developing domain-specific embeddings with more expressive language models trained in cybersecurity datasets.
- Addressing class imbalance using data augmentation techniques or advanced training strategies for low-resource entity types.
- Refine sequence modeling techniques to enhance the recognition of lengthy and structurally complex security entities.

REFERENCES

[1] P. Liu et al., "Multi-features based Semantic Augmentation Networks for Named Entity Recognition in Threat Intelligence," *arXiv preprint*, arXiv:2207.00232, 2022.

[2] Y. Lu et al., "Cybersecurity Named Entity Recognition Based on Word-level Enhancement and Multi-task Learning," in *Proc. 7th Int. Conf. Deep Learning Technol.*, 2023, pp. 71–78.

[3] M. Alam et al., "CyNER: A Python Library for Cybersecurity Named Entity Recognition," *arXiv preprint*, arXiv:2204.05754, 2022.

[4] P. Deka et al., "AttackER: Towards Enhancing Cyber-Attack Attribution with a Named Entity Recognition Dataset," *arXiv preprint*, arXiv:2408.05149, 2024.

[5] H. Gasmi, J. Laval, and A. Bouras, "LSTM Recurrent Neural Networks for Cybersecurity Named Entity Recognition," *arXiv preprint*, arXiv:2409.10521, 2024.

[6] Y. Qin et al., "Research on the Method of Network Security Entity Recognition Based on Deep Neural Network," *J. Nanjing Univ. (Natural Sciences)*, vol. 55, no. 1, pp. 29–40, 2019.

[7] Y. Xu, Y. Wang, T. Liu, et al., "Joint Segmentation and Named Entity Recognition Using Dual Decomposition in Chinese Discharge Summaries," *J. Am. Med. Inform. Assoc.*, vol. 21, no. e1, pp. e84–e92, 2014.

[8] Y. Zhang and J. Yang, "Chinese NER Using Lattice LSTM," *arXiv preprint*, arXiv:1805.02023, 2018.

[9] V. Mnih, N. Heess, and A. Graves, "Recurrent Models of Visual Attention," in *Adv. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 2204–2212.

[10] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," *arXiv preprint*, arXiv:1409.0473, 2014.

[11] W. Yin, H. Schütze, B. Xiang, et al., "AbCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs," *Trans. Assoc. Comput. Linguist.*, vol. 4, pp. 259–272, 2016.

[12] L. Wang, Z. Cao, G. de Melo, et al., "Relation Classification via Multi-Level Attention CNNs," in *Proc. 54th Annu. Meet. Assoc. Comput. Linguist.*, 2016, pp. 1298–1307.

[13] J. Devlin, M. W. Chang, K. Lee, et al., "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint*, arXiv:1810.04805, 2018.

[14] K. He, X. Zhang, S. Ren, et al., "Deep Residual Learning for Image Recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[15] J. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," in *Proc. 18th Int. Conf. Mach. Learn.*, 2001, pp. 282–289.

[16] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] N. Peng and M. Dredze, "Named Entity Recognition for Chinese Social Media with Jointly Trained Embeddings," in *Proc. 2015 Conf. Empirical Methods in Nat. Lang. Process.*, 2015, pp. 548–554.

[18] G. Lample, M. Ballesteros, S. Subramanian, et al., "Neural Architectures for Named Entity Recognition," *arXiv preprint*, arXiv:1603.01360, 2016.

[19] X. Ma and E. Hovy, "End-to-End Sequence Labeling via Bi-Directional LSTM-CNNs-CRF," *arXiv preprint*, arXiv:1603.01354, 2016.

[20] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," *arXiv preprint*, arXiv:1212.5701, 2012.