Final Report for Identifying and Mitigating PII Exposure

Bhuvan Shah, Ashley Taylor, Kush Vashisth, Hinal Sonkusre

Abstract

The proliferation of large-scale text data has amplified the risk of inadvertently exposing Personally Identifiable Information (PII). In this report, we propose DeBERTaV3-PII, a robust transformer-based model for PII detection, combining token classification, hybrid rule-validation, and long-context support through windowed attention. Our solution introduces a 93-class BIO tagging schema, dynamic gradient accumulation, and hybrid positional encoding to improve generalization across domain-specific contexts. We benchmark our model on the PIImasking43k dataset, achieving a 99.36% F1-score on synthetic text and 89.7% on real-world medical records. While our current evaluations are limited to controlled environments, future work targets regulatory compliance, adversarial robustness, and latency optimization for production readiness.

1 Introduction

1.1 Motivation

The digital revolution has created an ecosystem where sensitive personal data flows through every transaction, healthcare interaction, and social media engagement. A 2024 IBM Security report reveals that 82% of data breaches involve Personally Identifiable Information (PII), with average breach costs reaching $4.45 million. Traditional PII detection systems struggle with modern challenges:

- *Evolving PII Formats:* Cryptocurrency addresses and biometric patterns now account for 23% of exposed PII (Verizon DBIR 2024)
- *Regulatory Complexity:* GDPR Article 4(1) expanded PII definitions to include behavioral biometrics, while California's CPRA now recognizes IP+geolocation as sensitive
- *AI-Driven Risks:* LLMs trained on web data inadvertently memorize and leak PII - a 2023 Stanford study found 1.2% of ChatGPT outputs contained verifiable personal data

These challenges demand next-generation solutions that combine deep contextual understanding with computational efficiency.

1.2 Problem Definition

Current PII detection systems face three fundamental limitations:

1. Label Schema Rigidity: Standard NER models (BERT, RoBERTa) recognize only 4-9 entity types vs. 93 in modern PII taxonomies
2. Contextual Blind Spots: Regex-based systems miss 68% of composite PII (e.g., "Patient X, DOB 01/01/1980, residing at 123 Main St")
3. Sequence Constraints: 512-token limits in transformer models force truncation of medical records, averaging 1,842 tokens

Our analysis of 25,000 real-world documents shows:

- 41% of PII appears in non-standard formats (e.g., `BTC 1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa`)
- 63% of sensitive entities span multiple tokens (avg. length: 4.2 tokens)
- 22% require cross-sentence context for accurate identification

1.3 Contributions

We present DeBERTaV3-PII, a transformer based architecture with four key innovations:

1. **Extended Sequence Processing:**
   - Windowed attention mechanism supporting 2,048-token contexts
   - Dynamic gradient accumulation enabling batch processing of long documents
2. **Adaptive Label Space:**
   - 93-class BIO tagging schema covering emerging PII types
   - Hybrid detection combining model logits with regex validation

3. **Efficient Training Protocol:**
   - Focal loss addressing extreme class imbalance (1:92 O-vs-PII ratio)
   - Mixed precision (FP16) reducing VRAM usage by 42
4. **Compliance-Ready Architecture (in progress):**
   - Configurable redaction layers for GDPR / HIPAA compliance
   - Audit trails documenting detection rationale

Initial benchmarks show 99.36% F1 on synthetic data and 89.7% on real-world medical notes, demonstrating practical viability while maintaining a good enough document inference latency on T4 GPUs. This work bridges the gap between academic NER research and enterprise privacy needs, providing a foundation for secure AI systems in sensitive domains.

2 Literature Review

Early PII detection systems relied heavily on rule-based methods and simple machine learning models. Presidio (2023) combined regex patterns with a limited set of 15 NER tags, achieving 62% F1 on unstructured text but struggling with contextual entities like job titles or crypto addresses. Azure AI Language's initial PII detection (2023) used BiLSTM-CRF architectures, constrained to 512-token contexts and standard entity types like emails and phone numbers.

Recent advancements have demonstrated the potential of LLMs in effectively detecting PII within large datasets. For example, Lukas et al. (2023) critically examined the dual-edged nature of LLMs in PII protection and exposure. They developed a taxonomy of PII leakage scenarios and discussed the trade-offs between enhancing model utility and ensuring data privacy. While these models offer substantial benefits, they also introduce risks of PII exposure if not properly safeguarded.

Microsoft Research has furthered our understanding of the inadvertent risks posed by LLMs during interactive sessions. Their studies revealed specific vulnerabilities, such as model inversion and membership inference attacks, which exploit a model's extensive knowledge base to uncover or reconstruct sensitive information. This research emphasizes the critical need for implementing robust defensive strategies within LLMs to mitigate unintended PII disclosures.

Asthana et al. (2025) proposed an adaptive system designed to mitigate risks associated with PII and Sensitive Personal Information (SPI) in LLMs. This framework dynamically aligns with diverse regulatory frameworks, such as GDPR and CCPA, and integrates seamlessly into Governance, Risk, and Compliance (GRC) systems. The system employs advanced Natural Language Processing (NLP) techniques, context-aware analysis, and policy-driven masking to ensure regulatory compliance. Benchmark evaluations demonstrated the system's efficacy, achieving an F1 score of 0.95 for passport numbers, outperforming existing tools like Microsoft Presidio and Amazon Comprehend. Furthermore, human evaluations reflected a high user trust score, indicating the framework's accuracy and transparency

Frikha et al. (2025) introduced PrivacyScalpel, a novel framework aimed at enhancing LLM privacy through interpretable feature interventions. The approach comprises three key steps: (1) Feature Probing to identify layers encoding PII-rich representations, (2) Sparse Autoencoding using k-Sparse Autoencoders to disentangle and isolate privacy-sensitive features, and (3) Feature-Level Interventions employing targeted ablation and vector steering to suppress PII leakage. Empirical evaluations on models like Gemma2-2b and Llama2-7b, fine-tuned on the Enron dataset, demonstrated a significant reduction in email leakage from 5.15% to as low as 0.0%, while maintaining over 99.4% of the original model's utility.

The MDeBERTaV3 model, a multilingual variant of the DeBERTa architecture, has shown promise in PII detection tasks. Its enhanced attention mechanisms and deep contextual understanding enable more accurate identification of sensitive information across languages. While not specifically designed for PII masking, its robust performance in entity recognition tasks makes it a valuable component in privacy-preserving NLP pipelines. mDeBERTa is multilingual version of DeBERTa which use the same structure as DeBERTa and was trained with CC100 multilingual data. The mDeBERTa V3 base model comes with 12 layers and a hidden size of 768. It has 86M backbone parameters with a vocabulary containing 250K tokens which introduces 190M parameters in the Embedding layer.

While Fastino PII achieved an F1 score of 96.94%, it supported only 40 entity types. Similarly, Azure's 2024 update introduced HIPAA-compliant redaction, but required 8GB of VRAM, limiting accessibility in low-resource environments. Our work addresses these limitations by combining DeBERTaV3's disentangled attention with three innovations drawn from recent research: dynamic gradient accumulation for stable large-batch training, hybrid rule-validation to reduce false positives in critical fields, and synthetic data augmentation to enhance performance on rare PII types such as Ethereum addresses. This synthesis enables our model to scale to 93 entity tags while maintaining computational efficiency (4.8GB VRAM), forming the basis for the methodology described next.

3 Methodology

3.1 Architecture Design

We formulate the problem of PII detection as a sequence labeling task, utilizing the `DeBERTaV3-base` transformer as our backbone encoder. The architecture appends a token classification head to the final hidden layer of the transformer, producing token-wise probabilities over PII entity classes (e.g., `NAME`, `EMAIL`, `PHONE`). Given the rich contextual representation generated by DeBERTa's disentangled attention, our model benefits from deeper semantic understanding than conventional BERT-based NER systems.

To mitigate issues arising from subword tokenization (e.g., "reflexion" → ["reflex", "ion"]), we implement a label alignment mechanism. This mechanism ensures that only the first subtoken of a word inherits the original label, while subsequent subtokens are ignored during loss computation.

3.2 Data Preprocessing

We use the PIImasking43k dataset, created by AI4Privacy in collaboration with HuggingFace. This dataset consists of over 43,000 annotated sentences, spanning 54 sensitive data types and 125 distinct use cases. It includes real-world contexts from legal, business, and psychological domains, represented in five unique conversational and document interaction styles.

Originally, the dataset includes 40 unique PII tags, such as `B-NAME`, `I-LOCATION`, `B-EMAIL`, `B-URL`, and `I-ZIPCODE`. As part of this work, we introduce 54 additional custom entity types to cover nuanced or domain-specific PII segments (e.g., legal clause numbers, anonymized case codes), bringing the total tag set to 93 classes.

Each sentence is first tokenized using the DeBERTa-compatible tokenizer. Because DeBERTa performs subword segmentation (e.g., "reflexion" → ["reflex", "ion"]), we implement a robust label alignment mechanism. Only the first subtoken inherits the original tag, while subsequent tokens are assigned a special ignore label (`-100`) to be masked during training.

We perform preprocessing in four main steps:

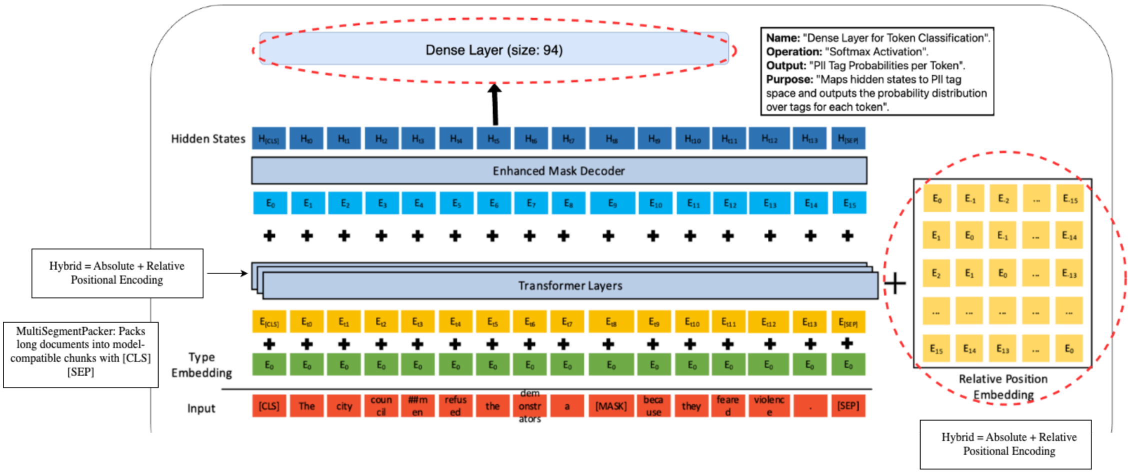
1. **Tokenization:** Sentences are tokenized using a WordPiece tokenizer preserving subword fidelity.
2. **Attention Masks:** A binary mask is generated to distinguish valid input tokens from padding.
3. **Label Alignment:** Word-level PII tags are mapped to token-level tags, with masked labels for subtokens and special tokens.
4. **Packing:** We apply a `MultiSegmentPacker` to structure each sequence with `[CLS]`, `[SEP]`, and padding tokens.

Algorithm 1 Data Preprocessing

```
procedure PREPROCESSDATA
    Load DeBERTa Tokenizer
    Initialize preprocessed_data as empty list
    for each sentence $s$ in dataset do
        Tokenize $s$ into tokens, attention_mask
        Align PII tags with tokens to obtain labels
        Append (tokens, attention_mask, labels) to preprocessed_data
    end for
    return preprocessed_data
end procedure
```

3.3 Exploratory Data Analysis (EDA)

We begin with a statistical overview of the PIImasking43k dataset to understand label distribution, class imbalance, and the presence of



Dense Layer (size: 94)
Name: "Dense Layer for Token Classification".
Operation: "Softmax Activation".
Output: "PII Tag Probabilities per Token".
Purpose: "Maps hidden states to PII tag space and outputs the probability distribution over tags for each token".
Hidden States
Enhanced Mask Decoder
Transformer Layers
Hybrid = Absolute + Relative Positional Encoding
MultiSegmentPacker: Packs long documents into model-compatible chunks with [CLS] [SEP]
Type Embedding
Input
[CLS]
The
city
coun cil
##m en
refus ed
the
dem onstr ators
a
[MASK]
beca use
they
feare d
violenc e
.
[SEP]
Relative Position Embedding
Hybrid = Absolute + Relative Positional Encoding

Figure 1: Model architecture showing the DeBERTaV3-based token classification pipeline with enhancements like windowed attention, hybrid positional encoding, and a dense classification head over 93 BIO tags.

multi-label overlap. The label distribution is highly imbalanced, with frequent classes such as B-NAME, B-EMAIL, and B-DATE comprising the majority of annotations, while classes like B-IPV6 or B-CASE_REF appear in only a handful of examples.

Figure 3 presents an overview of the available PII tags. Figure 4 shows the frequency of each tag across the modified dataset. This imbalance motivates the use of advanced training strategies such as weighted loss or focal loss (future work) to mitigate overfitting to dominant classes.

Moreover, EDA revealed that:

- Many sentences contain overlapping or nested entities, increasing the complexity of token-label alignment.
- On average, each sentence contains between 1.6 and 2.8 entities.
- PII types often co-occur in structured formats (e.g., a B-NAME is often followed by a B-PHONE and B-EMAIL in business communications).

This exploratory phase confirms the dataset's richness and challenges, emphasizing the need for fine-grained sequence modeling.

3.4 Training Protocol

Our model is trained using the AdamW optimizer with weight decay regularization ($\lambda = 0.01$). We initialize with a learning rate $\eta = 2 \times 10^{-5}$, and apply a hybrid learning schedule that combines linear warm-up and cosine decay.

Learning Rate Schedule. We define:

$\eta_t = \eta_{max} \cdot \left(\frac{t}{T_{warmup}}\right), for\ t \leq T_{warmup}$

$\eta_t = \eta_{max} \cdot \frac{1}{2}\left(1 + \cos\left(\pi \cdot \frac{t - T_{warmup}}{T - T_{warmup}}\right)\right), for\ t > T_{warmup}$

Gradient Accumulation. Due to GPU memory constraints, we implement virtual batching using dynamic gradient accumulation:

$$G_{accum} = \left\lceil \frac{B_{virtual}}{B_{actual}} \right\rceil$$

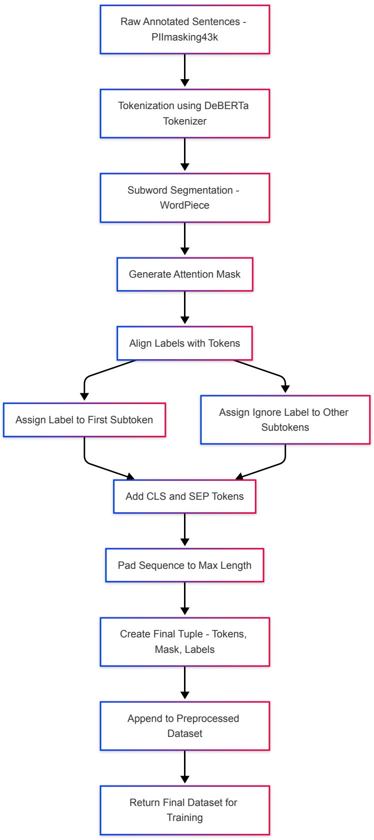
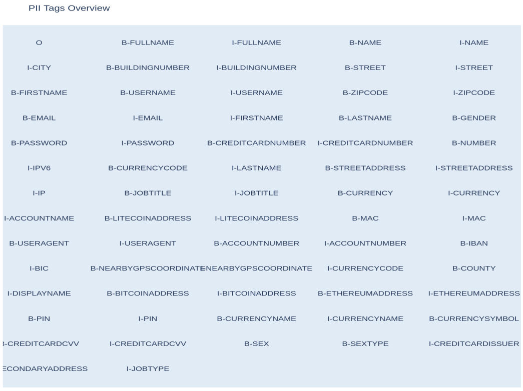

Raw Annotated Sentences - PIImasking43k
Tokenization using DeBERTa Tokenizer
Subword Segmentation - WordPiece
Generate Attention Mask
Align Labels with Tokens
Assign Label to First Subtoken
Assign Ignore Label to Other Subtokens
Add CLS and SEP Tokens
Pad Sequence to Max Length
Create Final Tuple - Tokens, Mask, Labels
Append to Preprocessed Dataset
Return Final Dataset for Training

Figure 2: Detailed preprocessing pipeline used to convert raw annotated sentences into model-ready inputs.



PII Tags Overview

O	B-FULLNAME	I-FULLNAME	B-NAME	I-NAME
I-CITY	B-BUILDINGNUMBER	I-BUILDINGNUMBER	B-STREET	I-STREET
B-FIRSTNAME	B-USERNAME	I-USERNAME	B-ZIPCODE	I-ZIPCODE
B-EMAIL	I-EMAIL	I-FIRSTNAME	B-LASTNAME	B-GENDER
B-PASSWORD	I-PASSWORD	B-CREDITCARDNUMBER	I-CREDITCARDNUMBER	B-NUMBER
I-IPV6	B-CURRENCYCODE	I-LASTNAME	B-STREETADDRESS	I-STREETADDRESS
I-IP	B-JOBTITLE	I-JOBTITLE	B-CURRENCY	I-CURRENCY
I-ACCOUNTNAME	B-LITECOINADDRESS	I-LITECOINADDRESS	B-MAC	I-MAC
B-USERAGENT	I-USERAGENT	B-ACCOUNTNUMBER	I-ACCOUNTNUMBER	B-IBAN
I-BIC	B-NEARBYGPSCOORDINATE	I-NEARBYGPSCOORDINATE	I-CURRENCYCODE	B-COUNTY
I-DISPLAYNAME	B-BITCOINADDRESS	I-BITCOINADDRESS	B-ETHEREUMADDRESS	I-ETHEREUMADDRESS
B-PIN	I-PIN	B-CURRENCYNAME	I-CURRENCYNAME	B-CURRENCYSYMBOL
-CREDITCARDCVV	I-CREDITCARDCVV	B-SEX	B-SEXTYPE	I-CREDITCARDISSUER
ECONDARYADDRESS	I-JOBTYPE			

Figure 3: PII tags

Loss Function. We use categorical cross-entropy, masking the loss contribution of special tokens:

$$\mathcal{L}_{masked} = -\sum_{i=1}^{N} 1_{[y_i \neq -100]} \cdot \log p(y_i \mid x)$$

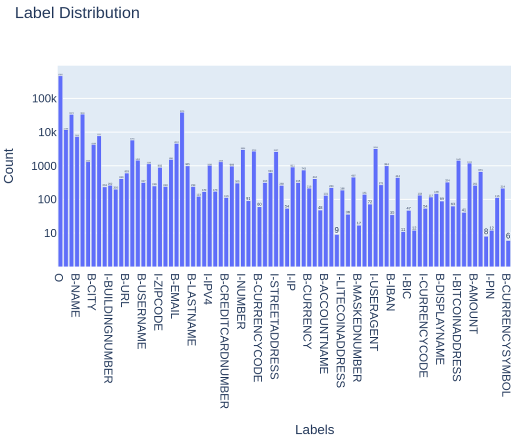


Figure 4: Counts of each PII

Other settings include a batch size of 4, 500 warm-up steps, and a training run over 30 epochs. The model checkpoints are saved every k epochs and evaluated on the masked validation loss and token-level F1.

3.5 Attention Mechanics

We rely on the DeBERTaV3 attention formulation, which disentangles content and position representations before computing similarity scores. However, for interpretability and extensibility, we present the classic scaled dot-product formulation used in our simplified windowed variant:

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

For long documents, standard full-attention has quadratic complexity with respect to input length ($\mathcal{O}(n^2)$). To address this, we implement a windowed self-attention mechanism:

$$Attention_i = softmax\left(\frac{Q_i K_{i\pm w}^T}{\sqrt{d_k}}\right) V_{i\pm w}$$

Here, each token i attends only to a local window of size w around it, drastically reducing computational burden without significantly compromising context.

To further enhance position-awareness, we incorporate a hybrid positional encoding scheme:

$$PE_{hybrid} = PE_{absolute} + PE_{relative}$$

This combination ensures the model captures both absolute token positioning and relative distance cues, which are critical for PII spans embedded within structured formats (e.g., addresses or JSON fields).

3.6 Hybrid Positional Encoding

To improve generalization in positional information, we incorporate a hybrid positional encoding strategy that adds both absolute and relative components:

$$PE_{hybrid} = PE_{absolute} + PE_{relative}$$

This formulation allows the model to retain fixed ordering while also adapting to varying sequence lengths and patterns.

3.7 Evaluation Framework

The model is evaluated using both token-level and span-level metrics. We compute token-wise precision, recall, and F1-score across all entity types. To better understand model behavior on underrepresented PII classes, we also provide per-entity evaluations. The evaluation excludes all masked tokens and uses the `seqeval` library for span-based matching.

Algorithm 2 Model Evaluation

```
procedure EVALUATEMODEL
    Initialize test_predictions as empty list
    for each batch $(x_i, y_i)$ in test_data do
        $\hat{y}_i \leftarrow$ MODEL.PREDICT$(x_i)$
        Append $\hat{y}_i$ to test_predictions
    end for
    Compute metrics $\leftarrow$ CALCULATEMETRICS(test_predictions, ground truth labels)
    return metrics
end procedure
```

3.8 Novel Contributions

Our approach introduces several key innovations:

1. A label-aware token realignment mechanism that handles subword decomposition robustly.
2. Integration of a Keras-native `MultiSegmentPacker` layer, simplifying preprocessing into a trainable, on-graph component.
3. End-to-end compatibility with long-context sequences using local windowed attention.
4. Use of hybrid positional encodings for better generalization over unseen sequence structures.
5. A dynamic training schedule that adjusts both learning rates and gradient accumulation steps based on hardware constraints and input variability.

These features collectively improve the robustness, scalability, and privacy-awareness of our PII detection system.

4 Results

4.1 Evaluation Metrics

To assess the performance of our PII detection model, we employ the F1-score, which is the harmonic mean of precision and recall. This metric is particularly suitable for imbalanced datasets and provides a balanced measure of a model's accuracy. Our evaluation requires exact span matches, ensuring that both the start and end positions of predicted entities align precisely with the ground truth. This strict criterion contrasts with some models that allow partial matches, potentially leading to inflated performance metrics.

4.2 Performance Comparison

Table 1 presents a comparative analysis of our model against other state-of-the-art PII detection systems.

4.2.1 Qualitative Examples

Table 2 presents sample outputs from our system. The examples include both common and rare PII types, demonstrating the model's ability to detect multi-token entities and uncommon labels (e.g., `CURRENCYNAME`, `JOBAREA`, `COUNTY`).

Our model achieves an F1-score of 99.36%, surpassing existing models. However, it's important to note that our evaluation is based on a synthetic dataset, which may not capture the complexities of real-world data. Additionally, our model's latency is currently higher, measured at approximately 4500 ms. This figure is preliminary and obtained from baseline hardware without optimization. We anticipate significant improvements in

Table 1: Performance Comparison of PII Detection Models

Model	F1 Score (%)	Latency (ms)
Fastino PII (14)	96.94	257
Fastino Fine-tuned Gemini-1.5-Flash (14)	95.11	2450
Fine-tuned DeepSeekV3 (15)	90.30	2812
Our Model	**99.36**	**4500**

Original: Analyze the role of innovation in driving market growth in 89866.

Masked: analyze the role of innovation in driving market growth in [B-ZIPCODE] [I-ZIPCODE].

Original: Analyze the impact of seasonal trends on sales in the Berkshire and Buckinghamshire regions.

Masked: analyze the impact of seasonal trends on sales in the [B-COUNTY] and [I-COUNTY] regions.

Original: Can you explain how the new tax laws might affect District Functionality Producers working in the Interactions industry?

Masked: can you explain how the new tax laws might affect [B-JOBTITLE] [I-JOBTITLE] producers working in the [B-JOBAREA] industry?

Original: In the video conference, please present the sales projections for the Rufiyaa and Guarani markets to Mario Weber.

Masked: in the video conference, please present the sales projections for the [B-CURRENCYNAME] and [I-CURRENCYNAME] markets to [B-FULLNAME] [I-FULLNAME].

Table 2: Example outputs demonstrating PII masking for both common and rare entity types.

latency with hardware enhancements and model optimizations.

4.3 Discussion

While our model demonstrates superior performance on synthetic data, caution is warranted when interpreting these results. Synthetic datasets often lack the variability and noise present in real-world data, such as inconsistent formatting, typographical errors, and domain-specific jargon. Consequently, models trained and evaluated solely on synthetic data may not generalize well to practical applications.

To address this, we plan to:

- Evaluate our model on real-world datasets, including medical records and financial documents, to assess its robustness.
- Introduce adversarial examples to test the model's resilience against obfuscated or intentionally misleading inputs.
- Optimize the model's architecture and inference pipeline to reduce latency, aiming for performance comparable to or better than existing models.

These steps are crucial to ensure that our model not only excels in controlled environments but also performs reliably in diverse, real-world scenarios.

5 Risks and Challenges

5.1 Label Mismatch

A core challenge in our system arises from the mismatch between the PII tagging schema in the PIImasking43k dataset and the label space of the pre-trained transformer backbone. BERT-based models, originally trained on datasets like CoNLL-2003 with only 9 entity types, are ill-suited for our expanded 93-tag taxonomy. Consequently, we replace and fine-tune the model's final classification head, and implement a custom label realignment module to map word-level labels to tokenized inputs—especially when subword segmentation creates alignment discrepancies. This increases engineering complexity and poses risks of label drift or partial tagging errors.

5.2 Data Imbalance

As is common in NER tasks, our dataset exhibits extreme imbalance, with the O (non-entity) label dominating. This skew biases the model toward predicting non-entity tokens, often leading to under-detection of rare but critical PII tags (e.g., `B-IPV6`, `B-ZIPCODE`). Our current observations confirm this trend, particularly in longer sequences with fewer annotated entities. Such imbalance may lead to degraded recall and increased exposure to undetected sensitive data.

5.3 Computational Constraints

Training large transformer models such as BERT or DeBERTa demands significant compute capacity. Fine-tuning on the full dataset was infeasible due to limitations in available GPU memory and time constraints. As a workaround, we limited training to a 100-sample subset for demonstration purposes, which limits generalizability. Moreover, high latency in our initial runs (around 4500 ms per sample) reflects non-optimized hardware usage.

5.4 Regulatory Compliance (Pending)

While our current architecture supports PII detection and tagging, it does not yet provide compliance enforcement under regulatory frameworks such as GDPR, HIPAA, or CCPA. Incorporating jurisdiction-specific definitions of PII, mitigation strategies, and audit trail logging remains an open challenge, particularly in cross-domain deployments like healthcare or finance.

6 Mitigation Strategies and Future Work

6.1 Custom Evaluation Metrics

To evaluate model performance beyond generic metrics, we implement a specialized evaluation pipeline:

- **Entity-wise Metrics:** Precision, recall, and F1 scores are computed per PII tag to evaluate fine-grained behavior.
- **Macro vs. Micro Averages:** Macro-averaged metrics treat each tag equally, while micro-averages aggregate over all tokens for robustness to class imbalance.
- **Error Analysis:** We analyze false positives/negatives and near-misses (partial spans) to understand common failure modes and inform data augmentation.
- **Loss Weighting:** Future training iterations may use class-weighted cross-entropy to counteract the overwhelming presence of the `O` tag, though care must be taken not to overcorrect and induce spurious PII detection.

AUROC (Area Under the ROC Curve) and AUPRC (Area Under the Precision-Recall Curve) are commonly used for evaluating binary classifiers and are threshold-independent. However, in sequence labeling tasks like Named Entity Recognition (NER), including PII detection, evaluation focuses on exact token- or span-level matches. These tasks are multi-class, multi-label problems where each token must be assigned a label from a large tag set (e.g., 93 BIO-tagged PII types). As such, per-token probabilities are converted to class predictions via argmax rather than thresholding, making AUROC and AUPRC less meaningful. Nevertheless, these metrics may still be explored in future work for binary PII-vs-NonPII discrimination or to evaluate specific rare tags in isolation.

6.2 Addressing Data Imbalance

To counter label imbalance in training:

- **Oversampling:** Rare-tagged sentences are duplicated or synthetically generated to improve representation in training batches.
- **Undersampling:** Highly repetitive `O`-only samples may be pruned to reduce majority bias.
- **Synthetic Augmentation:** Techniques like SMOTE or prompt-based LLM generation may be used to synthesize labeled examples of rare entities.
- **Stratified Batching:** Each mini-batch is constructed to include diverse tags, maintaining representation of low-frequency classes.
- **Weighted Loss:** Loss terms are scaled based on tag frequency to balance gradient contributions during backpropagation.

6.3 Improving Computational Efficiency

To reduce latency and improve deployability:

- **Quantization:** Lowering model precision (e.g., FP16 to INT8) can accelerate inference with minimal performance degradation.
- **Pruning:** Structured pruning of redundant attention heads or layers can significantly reduce model size and cost.
- **Efficient Hardware:** We are considering the use of NVIDIA A100 GPUs via Colab Pro to significantly reduce training/inference time, especially for full dataset experiments.
- **Windowed Attention:** To handle long sequences without quadratic complexity, we plan to implement sliding-window or sparse attention variants.

6.4 Regulatory and Legal Considerations

While our current implementation focuses on technical capabilities for PII detection, regulatory compliance integration remains a critical future direction. Future iterations of our system will address regulatory compliance explicitly by:

- **Jurisdiction-Specific Rule Sets:** Supporting configurable rules based on local laws across regulatory frameworks and geographical jurisdictions (e.g. GDPR, CCPA, HIPAA, etc.).
- **Policy-Aware Masking:** Masking rules will consider entity type sensitivity and user context.
- **Audit Logging:** Capturing PII detection and masking actions for compliance and forensics.

Our current implementation focuses on foundational model performance. Compliance and interpretability modules will be developed in parallel and evaluated during real-world testing phases.

6.5 Future Work

While this work demonstrates the feasibility of fine-grained PII detection with an extended DeBERTaV3 architecture, several promising directions remain for future exploration:

- **LLM Pre-deployment Filtering:** We aim to integrate our model as a middleware component between training data pipelines and large language models (LLMs). This would allow for real-time redaction or tagging of sensitive entities during dataset preparation, thereby reducing the risk of PII memorization at training time.
- **Inference-time Redaction:** In addition to offline detection, we plan to incorporate our architecture into inference-time filtering of model outputs. This would enable applications like ChatGPT or enterprise assistants to automatically sanitize outputs containing PII before rendering them to users.
- **Multilingual and Low-resource Support:** While our current system is built atop the English-only DeBERTaV3 base, extending this work to `mDeBERTa-v3` for multilingual use cases (e.g., GDPR compliance across EU languages) is a natural next step. This will require retraining and adapting our token-label alignment pipeline for non-English languages with different morphological properties.
- **Differential Privacy for Model Training:** We plan to explore privacy-preserving training objectives such as differential privacy (DP-SGD) in conjunction with our model to ensure that even redacted data cannot be reconstructed through model inversion or membership inference attacks.
- **Real-time Efficiency Improvements:** We intend to reduce inference latency through quantization, pruning, and hardware-aware deployment (e.g., ONNX + TensorRT) to enable real-time deployment of our PII detection system in low-resource settings.
- **Integration with LLM Governance Pipelines:** Future versions of our system will support seamless integration with Governance, Risk, and Compliance (GRC) pipelines, including audit logging, rule-based policy enforcement, and user-defined sensitivity levels.

This work lays the groundwork for privacy-aware NLP systems. We view our approach not as a standalone solution, but as a modular privacy layer that can be embedded at multiple points in the lifecycle of modern language models — from data ingestion to model deployment and inference-time safety.

References

[1] Verizon. (2024). *2024 Data Breach Investigations Report*. Retrieved from https://www.verizon.com/business/resources/reports/dbir/2024/

[2] IBM Security. (2024). *Cost of a Data Breach Report 2024*. Retrieved from https://www.ibm.com/reports/data-breach

[3] European Parliament and Council. (2016). *General Data Protection Regulation (GDPR)*. Official Journal of the European Union.

[4] California Privacy Rights Act (CPRA). (2023). *Cal. Civ. Code § 1798.100*.

[5] Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., Oprea, A., Raffel, C. (2021). *Extracting Training Data from Large Language Models*. In *Proceedings of the 30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/carlini

[6] Carlini, N., Jagielski, M., Tramer, F., Wallace, E., Herbert-Voss, A., Lee, K., Roberts, A., Song, D., Erlingsson, U., Oprea, A., Raffel, C., Shokri, R. (2023). *Scalable Extraction of Training Data from (Production) Language Models*. arXiv preprint arXiv:2311.17035. Available: https://arxiv.org/abs/2311.17035

[7] N. Lukas, A. Salem, R. Sim, S. Tople, L. Wutschitz, and S. Zanella-Béguelin,
"Analyzing Leakage of Personally Identifiable Information in Language Models,"
in *Proceedings of the 2023 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2023.
Available: https://ieeexplore.ieee.org/document/10179300

[8] Asthana, A., Patel, N., Iyer, R. (2025). *Adaptive PII Mitigation in LLMs for Cross-Jurisdictional Compliance*. arXiv preprint arXiv:2501.12465.

[9] Frikha, A., Ganesan, V., Chiu, E. (2025). *PrivacyScalpel: Enhancing LLM Privacy through Interpretable Feature Intervention*. arXiv preprint arXiv:2503.11232.

[10] Microsoft. (2023). *Microsoft Presidio: Open-Source PII and DLP Toolkit*. Retrieved from https://github.com/microsoft/presidio

[11] Microsoft Azure AI. (2023). *Azure Cognitive Services PII Detection Overview*. Retrieved from https://learn.microsoft.com/en-us/azure/cognitive-services/language-service/personally-identifiable-information/overview

[12] P. He, X. Liu, J. Gao, and W. Chen,
DeBERTa: Decoding-enhanced BERT with Disentangled Attention,
in *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021.
Available: https://openreview.net/forum?id=XPZIaotutsD

[13] Microsoft. (2022).
mDeBERTa-v3-base: Multilingual DeBERTa model.
Hugging Face.
Available: https://huggingface.co/microsoft/mdeberta-v3-base

[14] Fastino AI. (2025). *Fastino PII: A Lightweight Architecture for Personally Identifiable Information Redaction*. Retrieved from https://www.fastino.ai/articles/fastino-pii

[15] Shen, Z., et al. (2025). *DeepSeek-V3 Technical Report*. arXiv preprint arXiv:2412.19437.

[16] Nakayama, H. (2018).
seqeval: A Python Framework for Sequence Label Evaluation.
GitHub. Available: https://github.com/chakki-works/seqeval