scientific reports

Check for updates

OPEN

A hybrid rule-based NLP and machine learning approach for PII detection and anonymization in financial documents

Kushagra Mishra, Harsh Pagare & Kanhaiya Sharma✉

Safeguarding Personally Identifiable Information (PII) in financial documents is essential to prevent data breaches and maintain regulatory compliance. This research presents a scalable hybrid approach that integrates rule-based Natural Language Processing (NLP), Machine Learning (ML) approaches, and a custom Named Entity Recognition (NER) model for the accurate detection and anonymization of Personally Identifiable Information (PII). A varied and accurate synthetic dataset was created to replicate genuine financial document formats, enhancing model training and assessment. The model has attained a precision of 94.7%, a recall of 89.4%, an F1-score of 91.1%, and an overall accuracy of 89.4% on synthetic datasets. Additional validation on actual financial documents, such as audit reports and vendor bills, revealed a consistent performance with an accuracy of 93%. The study utilizes confusion matrices, ROC curves, and precision-recall curves to evaluate the model which further validates the model's capabilities and generalization ability. The suggested approach provides a robust and efficient solution for protecting sensitive information in operational financial contexts, markedly enhancing current methods for PII protection.

Keywords Machine learning, Natural language processing, Personally identifiable information, Data anonymization, Financial data security

Financial institutions deal with vast volumes of documents holding Personally Identifiable Information (PII), comprising Customer Names, Account Numbers, Email Addresses, and Government-issued IDs. It is vital to business and regulatory compliance, yet it represents significant privacy and security concerns and is a target of increasingly sophisticated cyberattacks. An example of such happened in July 2019, when Capital One reported a data breach that occurred through a misconfigured firewall that enabled a perpetrator to steal information from more than 100 million U.S. and 6 million Canadian credit card applications, including name, address, birthdates, Social Security numbers, and bank information. This depicts how a single data leak of PII can put millions at risk and cost more than $100 million to remediate and pay penalties. In order to maintain data utility and privacy, the institutions have focused on automated methods of detecting and anonymizing PII within financial documents. Institutions are required to meet intricate regulations (e.g., GDPR, CCPA) and safeguard against increasing cyberattacks[1]. This has accelerated the transition from conventional methods to advanced, scalable solutions. Recent studies like CPS-IoT-PPDNN underscore the increasing necessity for privacy-preserving and interpretable neural frameworks in managing sensitive data streams, emphasizing the need for information protection in critical systems[2]. Nonetheless, present methods have constraints that restrict their practicality. Rule-based and regex systems are straightforward to construct; however, they fail in unfamiliar formats and produce many false positives, making them unreliable for large unstructured financial text[3,4].

Initial machine-learning methods used unsupervised clustering on email datasets to identify clear indicators; however, they overlooked indirect signals, such as the combination of dates and locations, and required considerable data preprocessing[5]. Hybrid pipelines using TF-IDF, SMOTE, and regex-based classifiers such as Random Forest and SVM performed well on synthetic corpora[6]. However, their reliance on artificial training examples and manually crafted patterns constrains their real-world applicability. Deep learning and named entity recognition use data to acquire contextual knowledge to fill these gaps. The NERPII toolset anonymizes structured English documents using BERT and replacements generated using Faker Library[7], however, sentence meaning cannot be retained. A Korean adaptation of BERT/ELECTRA achieved an F1 score of 0.943[8],

Department of Computer Science & Engineering, Symbiosis Institute of Technology, Constituent of Symbiosis International (Deemed University), Pune, Maharashtra, India. ✉email: kanhaiya.sharma@sitpune.edu.in

though it required extensive computational resources and annotation efforts. Dialogue-focused NER models have achieved recall rates near to 100%, although at the cost of precision, highlighting the ongoing trade-off between identifying all PII instances and minimizing false positives[9]. Similar ML-driven architectures, including recurrent neural networks (RNNs) and LSTMs, have also demonstrated effectiveness in protecting sensitive information[10,11]. Comparing SVM and LSTM on synthetic email data showed that deeper networks have large computational costs despite good accuracy[12,13]. Ultimately, privacy-preserving frameworks address related issues but have different trade-offs.

Differential privacy introduces noise to safeguard individual privacy, but reduces the accuracy of detailed data relationships[14]. Federated learning maintains localized raw data, yet faces challenges related to synchronization and bandwidth[15,16]. Techniques such as k-anonymity and noise injection can be used on streaming and IoT data[17], although they may compromise data utility. Additionally, domain-specific evaluations identify additional challenges: random anonymization may distort semantic flow in medical records, as highlighted by Raj et al.[18], legal-text pipelines require consistent pseudonyms to preserve narrative coherence[19], and the comprehensive TAB benchmark indicates that numerous quasi-identifiers remain vulnerable under standard NER de-identification[20]. Despite notable progress, current methods for PII detection and anonymization approaches are limited by inflexible criteria, significant pre-processing, high computational costs, and poor privacy protection. Precision, recall, semantic coherence after anonymization, and unstructured text management are generally difficult for these systems. Manual annotation and difficulties hiding quasi-identifiers limit practical applicability, posing regulatory and reputational problems in finance. Thus, a scalable, integrated NLP-ML strategy is needed to detect and anonymize PII in context while keeping text utility and operational efficiency. This study offers a hybrid ML-NLP model to detect and anonymize financial document PII to fill these gaps. The proposed lightweight and scalable solution anonymizes and replaced important PII (e.g., Names, Social Security Numbers, Credit Cards, Phone Numbers, Emails, Addresses) with placeholders to preserve readability, inspired by efficient designs like GA-mADAM-IIoT[21]. Real-world reports (e.g., Bank of America Audit Report and vendor bills) and various synthetic datasets showed good precision (94.7%), recall (89.4%), F1-score (91.1%), and accuracy (89.4%), limiting false positives and negatives. Graphical confusion matrices and precision-recall curves show model robustness. The proposed technology allows secure, compliant management of sensitive financial records without compromising usefulness or confidentiality. In summary, financial institutions manage vast quantities of sensitive personal information, encountering considerable regulatory compliance and cybersecurity concerns. The constraints of current methodologies—such as inflexibility, elevated computational expenses, and inadequate semantic consistency—underscore the necessity for a sophisticated, scalable solution. This study proposes a hybrid NLP-ML framework designed for efficient and context-preserving detection and anonymization of PII, hence addressing existing gaps.

This study is structured as follows: Sect. 2 reviews related work and identifies research gaps. Section 3 details the proposed system and methodology. Sections 4 and 5 present results, limitations, and future directions. Section 6 summarizes key findings, and Sect. 7 shares the dataset and code availability. The next section contextualizes our approach within existing PII detection and anonymization methods.

Literature review analysis

A systematic evaluation of previous research on automated PII detection and anonymization was conducted to contextualize the proposed model, focusing on studies that address big, text-intensive datasets characteristic of financial records. Research Papers were included if they (a) developed or assessed comprehensive detection–redaction pipelines, (b) presented benchmark measures facilitating direct comparison, and (c) addressed advantages and constraints impacting regulatory compliance and operational scalability. Table 1 outlines the nineteen qualifying studies, including a summary of each study's content, fundamental technique, and primary strengths and weaknesses, thereby providing a definitive reference for evaluating the efficacy of the hybrid NER architecture presented in this paper.

The next section describes the technical framework designed to address previous research gaps in precision-recall balance, semantic consistency, computational efficiency, and handling unstructured data. The study starts with model selection and then creates and annotates realistic, diverse datasets for a solid foundation. Next, the hybrid rule-based and machine learning-driven spaCy NER architecture, training methods, and context-preserving automated anonymization process are detailed. These components comprise a scalable, high-precision system to improve PII detection and anonymization.

Methodology

The methodology section elaborates on the structured approach followed for the development and implementation of the system to identify and eliminate PII from financial records. This present study overcomes deficiencies in the available PII detection algorithms through an integration of machine learning and NLP techniques. The approach is designed to preserve the overall semantic integrity of the anonymized text while ensuring high recall and precision in identifying PII. The major steps involved in this work include Model Selection, Data Generation, Model Training, the Detection of PII, and Performance Evaluation. Each of these steps requires developing a robust, scalable system that would perform well in realistic settings. To effectively implement the proposed methodological framework, the selection of an optimal NLP model and framework was crucial. The subsequent section elaborates on the selection process, assessing different models based on particular operational and scalability criteria for financial document processing.

Research paper	Content summary	Methodology summary	Key strengths	Key weaknesses
J. Yang, X. Zhang, K. Liang, Y. Liu (2023)	ChatGLM2-6B LLM can detect PII in Chinese archival texts as accurately as a supervised BERT baseline, ensuring effective privacy protection without labeled data	Benchmarks GPT-style LLMs on a curated archival dataset; experiments with in-context learning prompts for PII tagging and redaction; evaluates precision/recall vs. fine-tuned BERT baselines	First systematic look at zero-/few-shot LLM PII protection; shows rapid deployment potential without heavy labelling	Evaluation confined to Chinese archival data—leaving generalizability and formal privacy guarantees unverified. LLM still incurs high inference cost and occasional hallucinations
Tomás J., Rasteiro D., Bernardino J. (2022)	Evaluates open-source tools (ARX, Amnesia) that apply k-anonymity, l-diversity and t-closeness to anonymize personal textual & tabular data	Practical anonymization of a public vaccine-tweet dataset; OSSpal scoring of functionality & privacy	ARX reliably anonymizes large datasets with fewer errors	Amnesia struggles on big data; some methods still allow re-identification.
Kulkarni P., Cauvery N.K. (2021)	C-PIIM automatically flags PII in massive unstructured email corpora.	Word2Vec/BoW topic modelling → Byte-mLSTM clustering on Enron emails to surface PII-rich clusters.	Works on unstructured text; outperforms hierarchical clustering.	Misses quasi-identifiers; heavy preprocessing; corpus-specific.
Jaikumar J. et al. (2023)	Hybrid ML + regex system labels PII tokens for privacy-preserving processing.	Synthetic Faker corpus → TF-IDF vectors → SVM/GNB/RF (RF ≈ 96% acc); SMOTE balances classes.	High accuracy and adaptable to new patterns.	Synthetic-only training may not generalise; regex brittle.
Mazzarino S. et al. (2023)	NERPII library detects PII with BERT and replaces it with realistic synthetic values.	BERT + Presidio NER → Faker synthesis keeps schema while masking identifiers.	Preserves data utility; easy CSV integration.	English-only; breaks inter-column semantics in complex tables.
Jang S. et al. (2024)	Korean-language NER model for personal-info extraction (F1 0.943).	Fine-tunes BERT/ELECTRA on 33 personal-info tags.	Addresses non-English detection gap; strong accuracy.	Requires large annotated corpus; limited cross-lingual transfer.
Mina M. et al. (2024)	Pipeline redacts PII in conversational agent logs; targets high recall (0.99).	Off-the-shelf NER + multi-method pseudonymization tuned to dialogue.	Rarely misses sensitive items.	Low precision (0.64) over-masks, reducing data utility.
Makhija A.K. (2020)	Compares DL (LSTM) vs. SVM for PII detection in unstructured emails.	Synthetic email dataset; evaluates precision, recall, F1.	SVM achieves 93%+ accuracy with simpler model.	Synthetic data limits external validity; DL models heavy.
Liu Y., Yang C.-Y., Yang J. (2021)	Graph-CNN + self-attention locates implicit sensitive terms beyond keyword lists.	Builds word-pair graph then attention scores for phrase salience.	Captures hidden PII phrases missed by rule systems.	High compute demand; needs quality training corpus.
Yang L. et al. (2024)	Generates differential-privacy synthetic datasets to protect PII while enabling analytics.	Histogram + Laplace noise → sample synthetic records; tests correlation retention.	Strong formal privacy guarantees; tunable budget.	Noise degrades fine-grained accuracy; computationally intensive.
Hathurusinghe R. et al. (2021)	Federated BERT-based NER extracts PII without centralising data.	Auto-annotated WikiPII → distributed training; evaluates noise robustness.	Preserves privacy via local training; good PII recall.	Label noise; FL synchronisation overhead.
Dash B. & Sharma P. (2022)	Survey of Federated Learning + secure multiparty computation for privacy-preserving FinTech analytics.	Conceptual review—no experiments.	Highlights state-of-the-art privacy tech for PII.	Notes heavy comm/compute cost; non-IID data issues.
Ren W. et al. (2021	End-to-end framework anonymizes IoT data streams, continuous telemetry & media for PII protection.	k-anon on streams, noise masking on metrics, YOLOv3 object blurring on images.	Covers heterogeneous IoT PII types; GDPR compliant.	Utility loss from noise; DL too heavy for edge devices.
Raj A. & D'Souza R. (2021	NER-driven anonymizer masks PII in medical records with multiple strategies.	Detects dates, names, places → tagging/generalisation/suppression/substitution.	Balances readability with privacy; flexible techniques.	Depends on NER quality; random substitution can distort semantics.
Garat D. & Wonsever D. (2022)	Automates anonymization of court documents while preserving narrative coherence.	Legal-tuned NER + agglomerative clustering assigns consistent pseudonyms.	High F1 for person names; maintains story flow.	Model performance drops on unseen legal styles; needs retraining.
Pilán I. et al. (2022)	TAB corpus & metrics benchmark text-anonymization efficacy against re-identification risk.	1 268 ECHR cases annotated; introduces risk-vs-utility scoring.	Public dataset spurs progress; holistic evaluation.	Legal-domain specific; quasi-IDs still difficult; annotation variance.
Satwik R. Kodandaram et al. (2021)[22]	Deep-NLP masking system identifies and replaces PII tokens (names, locations) with pseudo-data.	Tokenisation → embeddings → Transformer masking.	High precision on English data.	Relies on non-gold corpus; high compute; unknown multilingual support.

Continued

Research paper	Content summary	Methodology summary	Key strengths	Key weaknesses
Junhak Lee et al. (2022)[23]	Uses Shapley-explained tree models to assess how de-identification of Personal Information affects data utility.	Train RF/GB/etc. on pseudonymised healthcare data; Shapley ranks feature importance.	Interpretable insight into which PII removals hurt models.	Tree models heavy; insights may not generalise across datasets.
N. Bookert et al. (2023)[24]	Custom NER extracts granular privacy fields from IoMT policies to aid compliance.	Two-layer: multi-label theme classification → fine-grained NER.	Automates tedious policy-PII extraction.	Errors on poorly-structured policies; needs domain tuning.

Table 1. Literature review analysis of prior work.

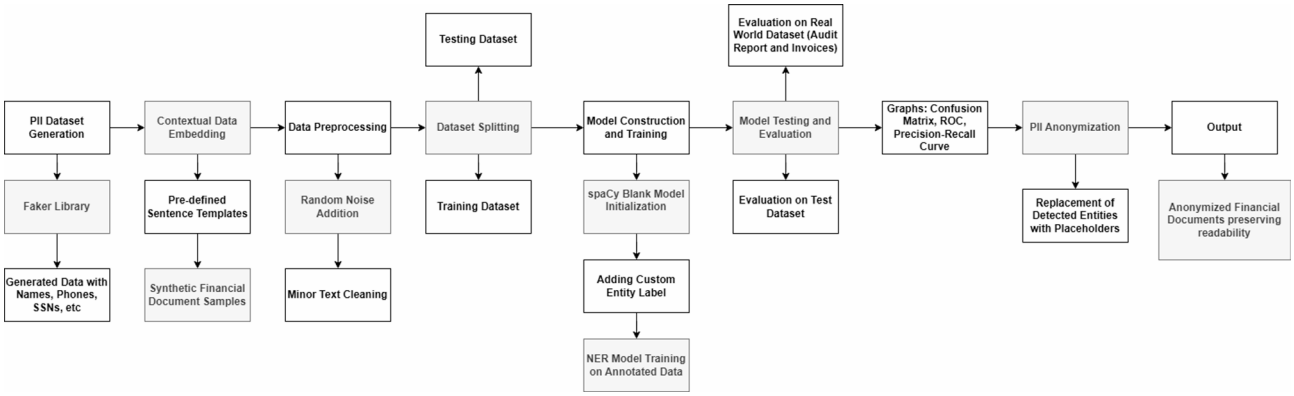

Fig. 1. System architecture for PII detection and anonymization.

Model and framework selection

Before constructing the custom NER Architecture and Pipeline, the study assessed various commercial NLP frameworks and pre-trained models for PII detection:

- SpaCy en_core_web_lg: A large English model trained on OntoNotes 5. Initial tests yielded only moderate recall for PII entity detection and inconsistent performance across categories.
- RoBERTa & ELECTRA: Transformer-based architectures offering strong language understanding, but in their default form both produced low PII recall and required extensive fine-tuning (and large GPU resources) to approach acceptable accuracy.
- FlairNLP: Sequence-tagging with contextual embeddings detected names, emails, and phone numbers but missed quasi-identifiers; achieving full coverage demanded significant retraining and hyperparameter search.
- StarPII: A specialized PII model combining rules and ML, yet limited to three PII entity types and exhibiting low overall F1-scores in preliminary evaluation.
- Microsoft Presidio: Offered easy anonymization pipelines, but its reliance on fixed recognizers proved brittle for varied financial PII formats.

Though transformer models (e.g. Hugging Face implementations of BERT, RoBERTa, ELECTRA) can achieve high accuracy on large corpora, the requirement for powerful GPUs, long fine-tuning cycles, and complex dependency stacks made them impractical for rapid iteration and deployment in this context.

By contrast, spaCy's lightweight blankmodel + NER pipeline provided:

- Rapid training and inference on commodity hardware.
- Simple extensibility to add new PII labels via ner.add_label().
- Minimal external dependencies and straightforward integration into existing Python code.
- Competitive performance once trained on the dataset.

A custom spaCy NER model was chosen for this study instead of transformer-based frameworks due to its speed, customization, and production readiness. The final framework choice, a customized spaCy NER pipeline, has the best performance, extensibility, and deployment speed, therefore the study now focuses on system architecture. The next part describes the architecture that integrates data creation, preprocessing, model training, inference, and assessment.

System architecture

Figure 1 displays the optimum system architecture for the proposed PII detection and anonymization approach, showing a systematic workflow from synthetic data generation to anonymized output. Using the Faker Library, authentic personally identifiable information (PII) such names, emails, addresses, phone numbers, credit card numbers, and social security numbers is generated. During Contextual Data Embedding, financial document

templates contain these PII components to reproduce audit reports, compliance notifications, and invoices. To better simulate real-world document anomalies, Data Preprocessing adds minor random noise like punctuation removal and text normalization. To ensure model generalization and impartiality, the dataset is split into a Training Dataset and a Testing Dataset after preprocessing.

In the Model Construction and Training phase, a blank spaCy Named Entity Recognition (NER) model is initialized and customized with entity labels for multiple PII categories. This model learns rule-based patterns and contextual representations of PII data from the annotated synthetic training dataset. In Model Testing and Evaluation, the trained model is tested on the synthetic Testing Dataset. The Evaluation metrics include Precision, Recall, F1-Score, and Accuracy. Inorder to visualize the performance, Confusion Matrices, ROC Curves, and Precision-Recall Curves are being created.

Real-world financial documents like Audit Reports and Vendor Invoices are used to test the Model to prove its reliability and accuracy on actual financial reports. PII Anonymization accurately replaces recognized PII components with placeholder tokens, preserving the original text's grammar and context. The approach creates Anonymized Financial Documents that are readable for operational or analytical reasons while protecting sensitive data. This comprehensive pipeline - including synthetic data creation, contextual embedding, noise-based preprocessing, rigorous training, real-world validation, and controlled anonymization - provides scalable, efficient, and regulatory-compliant financial data protection. The proposed system architecture integrates data preparation, model training, PII detection, anonymization, and performance evaluation.

This methodology facilitates sensitive data anonymization and identification with retention of context, improving operational effectiveness and regulatory adherence. Once the architecture pipeline has been set, the next step is to offer representative data. The next section now focuses on how a synthetic yet realistic dataset can be created and curated.

Creation of training and testing dataset

A good quality of training and testing dataset is very important to boost performance. Because of the data being sensitive, the proposed method relies on the creation of a synthetic dataset that resembles actual financial records and contains various types of PII. Using the Faker library authors can actually reproduce realistic PII data with a variety of complex samples from which the model can learn. This will also contribute to enhancing the learning process of the model by exposing it to different PII patterns and scenarios.

The creation of the dataset first needs to involve the development of various types of PII through the Faker module for a dataset that is deep and varied. In Faker, these particular data types will have special functions in generating personally identifiable information. Specific data types generated include:

- Authors generate full names, including all genders and cultural backgrounds, via the fake.name() function. Again, these will include a mix of common and unusual names. This diversity in variety helps train the model to pick out the names in different contexts and guises.
- Wrote a custom method generate-phone-number() providing generated phone numbers in an assortment of international formats using Faker.
- Various formats of email addresses can be generated through the fake.email() function. This reflects real-life email addresses that would come in different domains and naming protocols. Conclusively, the model is then set to understand, through practice with such variants, the email addresses in many various settings.
- The fake.address() function would return a realistic address with the street name, city, state, and valid postal code, the selection of which depends on the region. Single-line addresses and multi-line addresses: The goal here is to give the model an opportunity to repeat real-life address structures in as many variations as possible.
- The fake.ssn() function returns multiple formats for SSNs to account for regional or national differences. By including these variants, the model is able to detect SSNs in various formats with and without hyphens, such as "123-45-6789".
- The fake.credit-card-full() function generates the complete credit card information with a card number, expiration date, and security code. This gives realistic examples of what inputs could be on real paper by emulating different card types like Visa and Mastercard.
- fake.company() returns a list of realistic company names across various industries. This will emulate the variety that exists within real organizations. Examples will range from common to unique. The model needs to learn about the different presentations of company names within different contexts of documents.
- To build website URLs, authors utilze fake.url(). Their structure is different, and domain type differs (.com,.org,.net), appearing as in real financial documents. By including different URL structures, the model is sure to learn how to identify a URL regardless of the form it takes.

Although the Faker library generates synthetic PII quickly and efficiently, its default settings are culturally biased, especially in name, address, and identification information representation. The default name pool is mostly Western, therefore address formats may be U.S.-centric. Unaddressed biases could limit the model's applicability to non-Western or multicultural datasets. To avoid this, the study randomly assigned formats across areas and included multi-line or single-line addresses. A custom formatting method (generate_phone_number()) created phone numbers using randomized international codes (e.g., +91, +44, +61), and SSNs were synthesized in numerous valid formats (with and without hyphens) to represent regional variances. Faker's multilingual capabilities was selectively activated during controlled experiments to demonstrate generalization across cultural data points, although financial reporting rules kept English as the primary text language. Additionally, randomized phrase templates were used to embed the generated data in varied linguistic and contextual contexts, reducing overfitting to document patterns. To do this, authors created phrase templates that resemble financial

documents, reports, tax filings, compliance notices, transaction confirmations, etc. Additionally, these models have placeholders for several PII types, such as {name}, {company}, {email}, and {ssn}.

These templates are randomized with PII data to create sentences that should match how PII would appear in the documents. For example:

- Template: "We are writing to confirm that the payment against the invoice INV-12345 issued by {company} has been successfully completed. The payment was facilitated by {name} using a credit card, ending in {credit_card} ".
- Generated Sentence: "Payment against XYZ Corp invoice no. INV-12345 is successfully made. The payment is facilitated by John Doe by using the credit card ending with 1234-5678-9012".

The model would learn how specific PII is discovered and how it occurs most in different text settings, allowing it to generalize across broad applications. The authors used randomized techniques to add variation and enhance the model. One is random full stop removal from output text based on a pre-specified probability, such as 30%. By integrating formatting or typing errors from real-world data, this makes the model more robust and adaptable. The dataset was created using randomization, sentence patterns, and expression variants of common word combinations. For instance, "payment" can signify "remittance" or "settlement." Common phrases like "facilitated by," are reinterpreted as "processed by" or "handled by." Because it does not grow inflexible in detection patterns, the model can accept diverse phrasing of the same events in real-life applications where language usage is likely to be variable. To further validate the qualitative realism of the dataset, the design of the synthetic templates was grounded in close observation of the structure, tone, and content of real-world financial documents. Templates were crafted with direct reference to authentic audit reports, tax filings, account confirmations, and compliance notices, ensuring alignment with the formal language, multi-clause structure, and hierarchical formatting commonly found in such documents. Domain-specific expressions such as "TIN verification," "regulatory filings," or "asset de-recognition" were intentionally embedded to reflect the technical vocabulary typical in professional financial narratives. Additionally, the position of PII was varied—appearing in headers, body text, and even footnote-style disclosures—closely mimicking the unpredictable yet context-rich placements observed in enterprise reports. This meticulous alignment enhances the contextual believability of the dataset, enabling the model to generalize better when applied to actual financial environments.

Authors help the model recognize PII and understand text context. It can distinguish between PII data and false positives like "numbers" from non-PII business data with significant learning. An extensive variety of templates and contexts improves model accuracy and efficiency on varied real-world datasets. These changes expose the model to more PII representations, boosting its robustness in real-world, cross-cultural financial datasets. The final datasets are constituted of 45,030 synthetic entries in separate CSV files for ease of use in subsequent training and testing stages (Information on Dataset Availability is present in Section VI). Each dataset's typical row contains PII and a statement designed to contextualize the data. The authors allow structured data storage that integrates into the model training and testing pipeline to make learning efficient and successful. Figure 2 illustrates the complete workflow for generating a synthetic dataset for training and testing the PII detection model. The process begins with data collection and raw input of financial documents, followed by preprocessing and a dedicated data cleaning phase to ensure consistency and remove noise. Next, sentence templates are created to emulate real-world financial text structures. Data augmentation is performed by initializing the Faker library, specifying the number of PII instances to be generated, and synthetically creating various PII types, including names, phone numbers, email addresses, credit card numbers, SSNs, company names, and URLs. These synthetic PII values are then applied into sentences using the generated templates. Additional robustness is incorporated by introducing noise through random removal of full stops in sentences. Finally, the synthetic sentences are compiled and saved into structured datasets, ready for model training and testing. Each block in the figure represents a critical stage in ensuring that the dataset is realistic, diverse, and contextually rich for effective PII detection.

Inorder to summary this segment, the carefully constructed synthetic dataset plays a vital role for the training and validation of the proposed model. Through the support of simulating realistic and varied financial document scenarios - varied formatting, cultural representation, and complex contextual embeddings - the dataset ensures the robustness and real-world readiness of the model. After the creation of synthetic datasets, accurate PII location detection and annotation are important in correctly training and validating the model. Therefore, the upcoming segment explains the procedure for annotation, detailing on how accurate PII are annotated to ensure optimal learning and test assessment.

Annotation of true PII data position in training and testing dataset

For building an efficient detection model, the precise annotation of the locations of PII in both the training and testing datasets are important. It consists of locating the exact starting and ending of each instance of PII within the text and noting them. In this way, the authors are feeding the model with real-life ground truth data through which it learns on how to identify and mask sensitive information according to various contexts. This section provides a detailed explanation of how the annotation is done and the precautions taken to ensure the dataset is well prepared for training and evaluation. The aim of this work was to train a model that can identify PII and understand where it fits contextually in a document.

The overall annotation process involves a number of critical stages that make a dataset more reliable and correct. It includes loading the dataset, defining the annotation function, thereby locating and recording the PII points, applying the annotations, and saving the annotated dataset for use in the training and testing phase. In inclusion, mathematical formulations are quite crucial in this process since they ensure accurate identification and marking of PII parts.

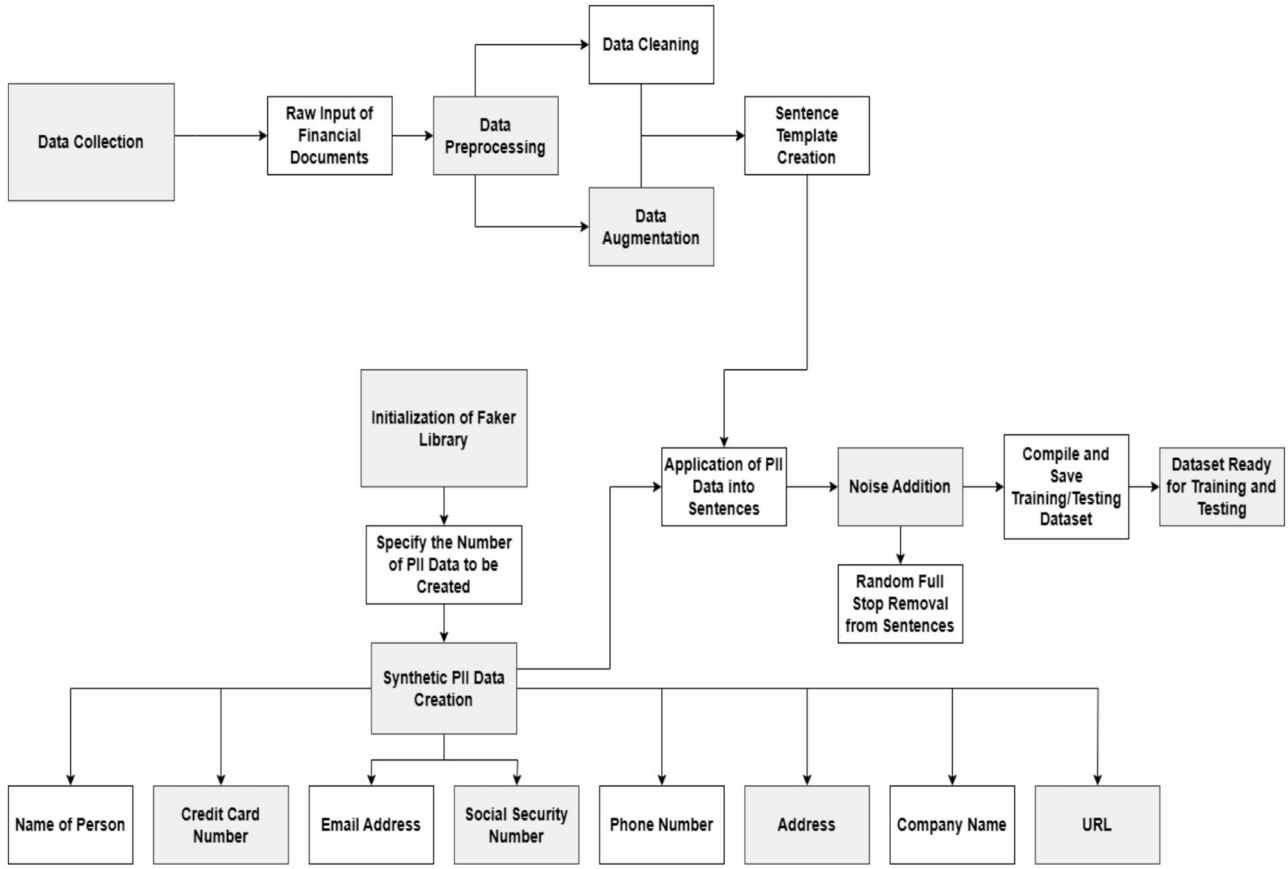

Fig. 2. Workflow for creating the synthetic training and testing dataset.

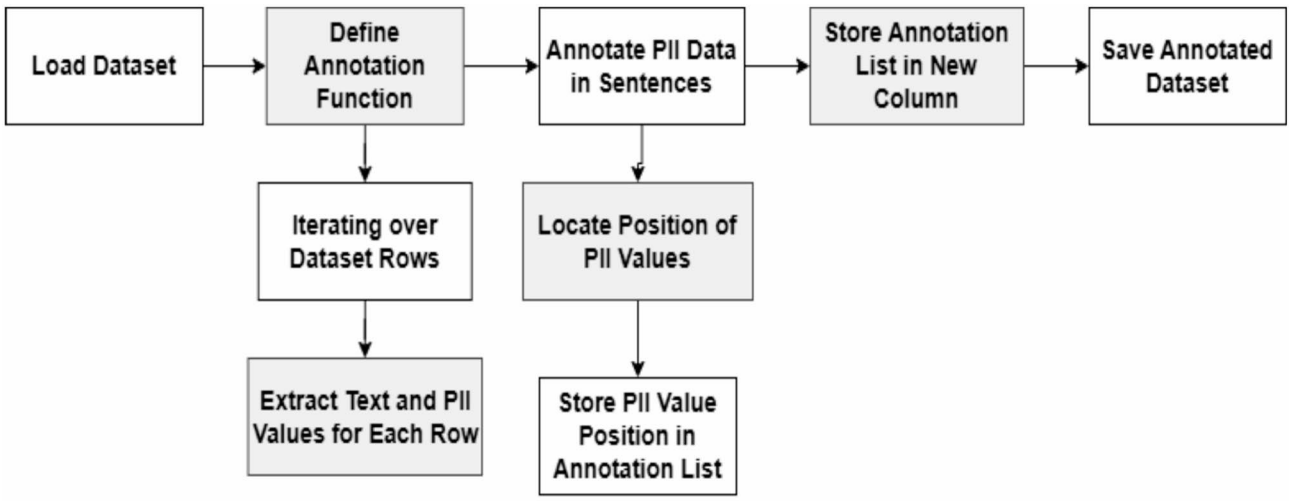

Fig. 3. Workflow for annotating True PII positions in training dataset.

Figure 3 visualizes the structured workflow used for annotating the true positions of PII entities in the datasets. It starts by loading the dataset containing the financial text samples with synthetic PII embedded. Then, an annotation function is defined to search for PII values within each sentence. The dataset rows are iterated sequentially, and for each sentence, the text and associated PII values are extracted. Using regular expressions, the start and end character positions of PII entities are precisely located. These identified spans, along with their entity types (e.g., Name, Email, SSN), are stored systematically in an annotation list. The list is subsequently added as a new column to the dataset. Finally, the annotated dataset is saved for downstream model training and

evaluation. This block-wise approach, captured in Fig. 3, ensures consistent, high-quality labeling essential for building an effective PII detection model.

The script starts the annotation process by loading a synthetic dataset that contains different forms of PII embedded in the text samples. For ease of use, this dataset will be loaded from a CSV file into a DataFrame using the pandas package. This is the crucial stage, having provided the raw text data to identify and label PII elements up until this point, setting the stage for annotation.

The authors will first load the dataset and define the function *annotate_pii*. This function will search for an exact match of PII values supplied in each row within the text and annotate their positions. This function iterates through each kind of PII, finds all its exact matches in the text with the help of regular expressions, and outlines their start and end positions. Because this task uses regular expression matching, the mathematical intuition behind this step will be:

$$Match(T, P) = \{(i, j) \mid T[i : j] = P\} \tag{1}$$

.

Here 'T' is the text sample and 'P' is the PII value to be found, and (i, j) will indicate the start and ending indices where 'P' appears in 'T.'

This feature is crucial to the annotation process, as it systematically identifies all instances of personally identifiable information within the text and ensures the annotations made are both complete and accurate.

For this, after the implementation of the annotation functionality, authors support the finding and noting of the exact position of each PII type inside the text. This functionality retrieves the start and end indices of every discovered PII instance. In order to train the model on what to focus its attention on while identifying PII, this exact labeling is necessary. With that in mind, an annotation for each instance could be written as:

$$Annotation(T, P) = \{(i, j, Type)\} \tag{2}$$

.

Here, 'i' and 'j' represent the starting and ending index of PII and 'Type' denotes the category of PII (e.g., 'name', 'email').

The span length L of each PII can be computed to more fully understand the extent of each element of PII:

$$L = j - i \tag{3}$$

.

Here, 'L' is the length of PII in the text, whereas 'i' and 'j' are the start and end index, respectively.

Once the positions of PII in the text have been precomputed and remembered, each sample in the dataset locally runs through the function annotate_pii. During this step, the annotations of each text example are appended to a new column in the DataFrame. The annotations are represented as a set:

$$A = \{(i_k, j_k, Type_k) \mid k = 1,2, \ldots, N\} \tag{4}$$

.

where 'N' is the total number of PII elements in the text. $(i_k, j_k, Type_k)$ denotes the start index, end index and type of each PII respectively.

The final stage of annotation is to save the annotated dataset into a new CSV file. Apart from this dataset, a new feature is the column 'True Predictions,' which enumerates the exact positions of PIIs in every text sample. This annotated data provides information with exact specifications as to where the model is supposed to identify and anonymize PII; therefore, it acts as a ground truth for the model's testing and training.

Figure 3 shows the block diagram and information on the systematic process applied to mark the places of PII within the training and testing dataset. By loading the dataset, constructing an annotation function, and using the function to correctly label PII places, the authors generate a complete dataset, which is an extremely important tool in training and testing the PII detection model. It achieves this through mathematical formulations and regular expressions, forming the bedrock of a strong and reliable PII detection system by identifying and annotating each and every PII element with high accuracy.

With a meticulously annotated dataset established, the study now advance's to model training, the fundamental element of the proposed PII detection solution. The subsequent section delineates the procedures for training the bespoke Named Entity Recognition (NER) model utilizing annotated data to precisely identify and categorize diverse types of Personally Identifiable Information (PII).

Training the model

The proposed PII detection technology requires model training. Authors utilize the spaCy library's Named Entity Recognition (NER) model to recognize and classify text entities. The NER model learns to detect names, credit card numbers, and email addresses from an annotated dataset. Configure the NER pipeline, prepare the data, then run iterative training loops to maximize model performance. NER categorises named entities in unstructured text. Here, PII includes "name," "credit_card," "email," etc. The NER model learns from annotated text to identify entities. It trains model parameters to minimize the differences between its predictions and the actual annotated entities to improve PII detection in new, untested data. Figure 4 shows in detail the internal mechanisms of the NER model. It does that in graphical steps: text tokenizing, feature extraction, and finally, token classification into entity types. As a result of this standardized process, the model will be able to read the text with full efficiency to identify PII. It is important to understand how a model would transform unstructured

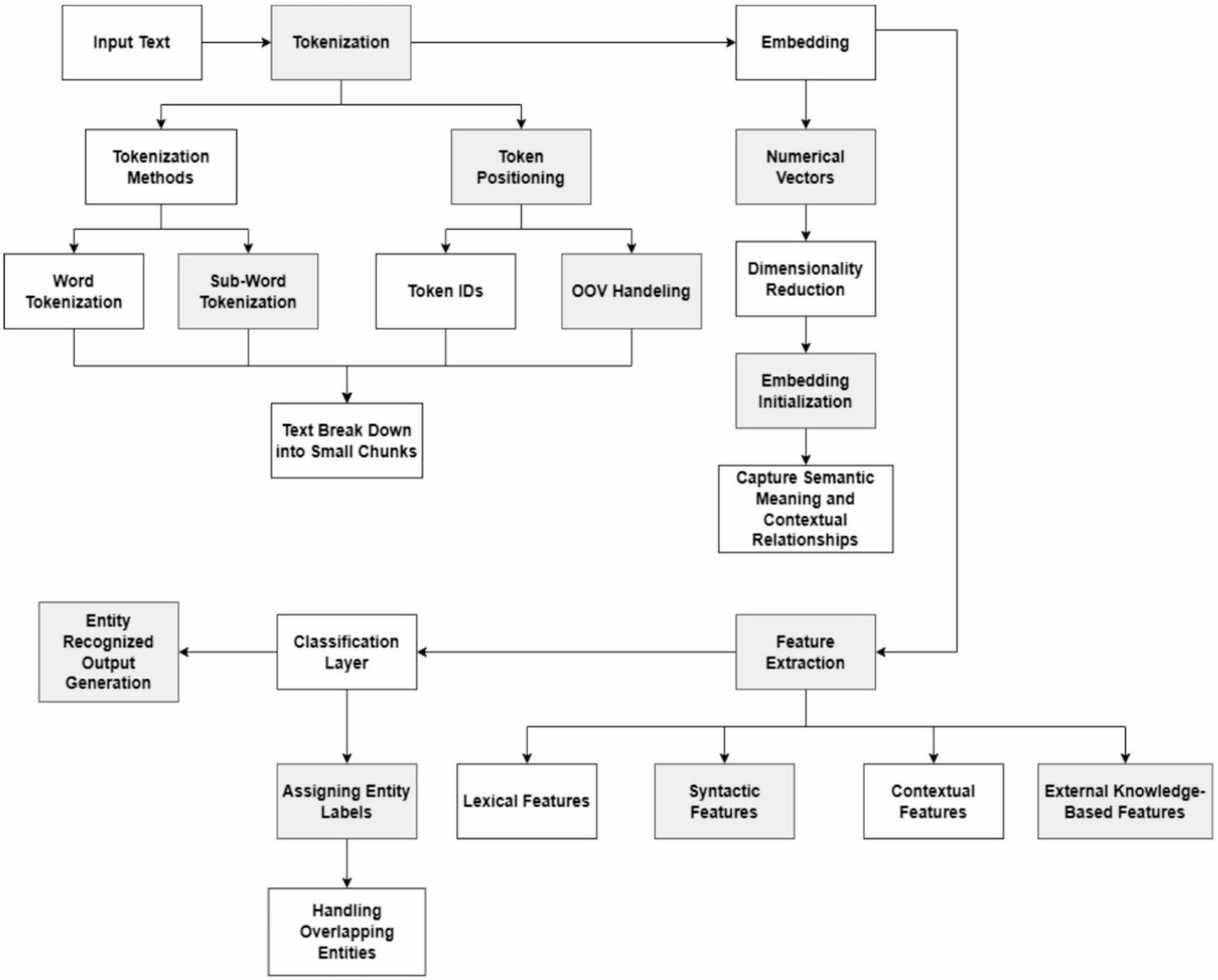

Fig. 4. Internal workflow of the named entity recognition (NER) model.

text data into a structured format that highlights sensitive information. The workflow begins with the input texts, which first are tokenized into smaller chunks using tokenization methods such as word-level and subword-level tokenization. Each token is then assigned a position and a token ID, when handling out-of-vocabulary (OOV) tokens to ensure robustness. After that these tokens are embedded into numerical vectors by embedding initialization, with dimensionality reduction applied to optimize computation. The model can then capture semantic meaning and contextual relationships amongst tokens, followed by feature extraction across lexical, syntactic, contextual, and external knowledge-based dimensions. Finally, entity labels are assigned to each token by a classification layer, it handles overlapping entities before producing the recognized entity output. This multiple-step workflow ensures that the NER model efficiently processes complex financial text to accurately identify PII elements.

Because of this, Tokenization is therefore, the first step of the NER model, by which the input text is divided into discrete tokens. Words or punctuation are normally picked up by these but can be anything based on the needs. This breaks the raw text down into small pieces that the model can work with. This is important because tokenization keeps the structure and order of the words while letting the model look at each part of the text separately. This step makes sure that the material is made in a way that the model can understand and use, which sets the stage for further analysis. Once the tokens are extracted, each token is then embedded into a numerical vector. These embeddings represent representations of tokens in a multidimensional vector space regarding their semantic meaning. The approach can capture language subtlety by grouping tokens with comparable meanings or contextual applications. Within financial documents, words like "credit" and "card" contain embedded meanings that display their relationship. Embeddings provides rich semantic inputs for accurate learning. Besides tokenization and embeddings, entity alignment is crucial to model validity. It matches tokens like name, email, etc. that the model defines as specific entities with the source text entity. If several entities take more than one token, the model must label all tokens the same. For example, "John Doe" is one full name. By overlapping or joining entities, this alignment method makes it easier to find entities in complex texts like legal papers and financial statements. During the training phase, there are also times when entities are not aligned or are in

conflict. This happens most often when there is a difference between what the model predicts and how the input text is tokenized. The workflow includes a way to find and fix misaligned entities by either realigning them or ignoring them in the training data. Addressing irregularities is crucial in building a stronger, more adaptive model that generalizes a variety of noisy real-world data well. Such alignment techniques make the NER model strong in identifying and classifying PII in document format. After that comes the extraction via the model of features from the token embeddings, conveying context about every token. Capturing the meaning added by tokens involves observing patterns and relations between them. The model uses such properties to identify linguistic patterns—such as names or addresses—that indicate the presence of an entity. This reminds us that successful entity recognition heavily depends on whether the model is able to distinguish which tokens belong to an entity and which do not by extracting such properties.

After feature extraction, the classification layer runs the extracted features. This is the layer in which each token is assigned a tag identifying whether or not it belongs to an entity and, in such a case, of what kind, for example, 'B-NAME' for the beginning of a name or 'I-EMAIL' for the body of an email address. With the classification performed on features extracted in previous steps, the model will, therefore, have the ability to distinguish between different kinds of PII. Let's classify a token's position in an entity using the BIO tagging scheme: Begin, Inside, Outside. This classification procedure lets the model generate structured data on the entities in the text. The last block gives the output, with entities identified by the model and their corresponding labels. This output provides the entity's detected names, credit card numbers, or email addresses- and their locations in the text. The NER model efficiently facilitates tasks like information extraction and anonymization through the annotation of recognized entities in input text.

Now, let us go into the details of the training process. Figure 5 describes visually how the annotated dataset is transformed into a trained model. This organized technique provides a methodical training of a model, which gradually improves its ability to find and classify PII over successive rounds of training and optimization. As depicted in Fig. 5, the training workflow initiates by loading the annotated dataset and preprocessing the data to extract text and associated PII labels. Entities are parsed by recording their start and end positions, followed by sorting and alignment to ensure consistency. Overlapping entities are merged systematically to avoid redundancy. A custom spaCy model is initialized with a blank pipeline, to which the NER component is added. The training examples are formatted using spaCy's Example class, ensuring accurate text-entity alignment. Model training is conducted over multiple iterations using a compounding minibatch strategy and dropout for regularization. Misaligned entities are handled gracefully by either adjusting spans or skipping problematic examples. After each batch update, the model's parameters are refined, and the final trained model is saved for downstream testing and deployment.

Preprocessing starts by importing the annotated dataset from the CSV file, which contains samples of texts with their corresponding PII annotations. After that, the data is structured in a consumable format for spaCy. In particular, each text sample is aligned with a dictionary of entities; each entity is defined by its label, e.g., "name," and start and end indices in the text. During this step, the main data structure to be used for model training is set up.

Mathematically, the training data 'D' may be represented as:

$$D = \{(T_k, E_k) \,|k = 1,2, \ldots , N\} \tag{5}$$

.

Sometimes, the parts of PII in a text can be overlapping, which creates problems for the trainees. To handle this, the authors created a function that would merge overlapping items as an item. Consistent with merging the items along the list of entities, the function would combine those whose index overlaps into one span.

Intuition: Assume two overlapping entities

$$(i_1, j_1)\ and\ (i_2, j_2)\ where\, i_2 \leq\ j_1$$

.

In this case, the merged entity will be defined as:

$$Merged\ Entity = (i_1, \max\,(j_1, j_2)) \tag{6}$$

…….

Then, the authors begin by creating an empty spaCy model for the English language with *spacy.blank("en")*. The main component to this pipeline is added: NER itself, which is responsible for learning to identify PII. NER is then configured to have the model ready to process the incoming training data and change their inner parameters and weights during the training. After combining the overlapping entities, the text and related entities are to be transformed into the spaCy format that the model is going to need for training. Now, the authors utilize an 'Example' class to create training examples in a format that aligns the text with the entities in a way spaCy's NER can process. This includes an assurance from the object spans that lie within the text-align and follow the intended structure. Once the training data is prepared, the labels or entity types are to be added to the NER component. To add the label of each entity in the training samples, the author use *ner.add_label*. Basically, it means telling the model what kind of entities- like "name," "credit_card," "email," etc.- one wants to learn during training to detect. In this study, the NER component was trained over 20 full passes of the annotated corpus (*iterations = 20*). Training began with a call to *nlp.begin_training()*, which instantiates SpaCy's Adam optimizer configured with its default decoupled weight-decay schedule. On each iteration, Example objects were drawn into mini-batches via *spacy.util.minibatch*, employing a compounding batch-size strategy that grew from 4 to

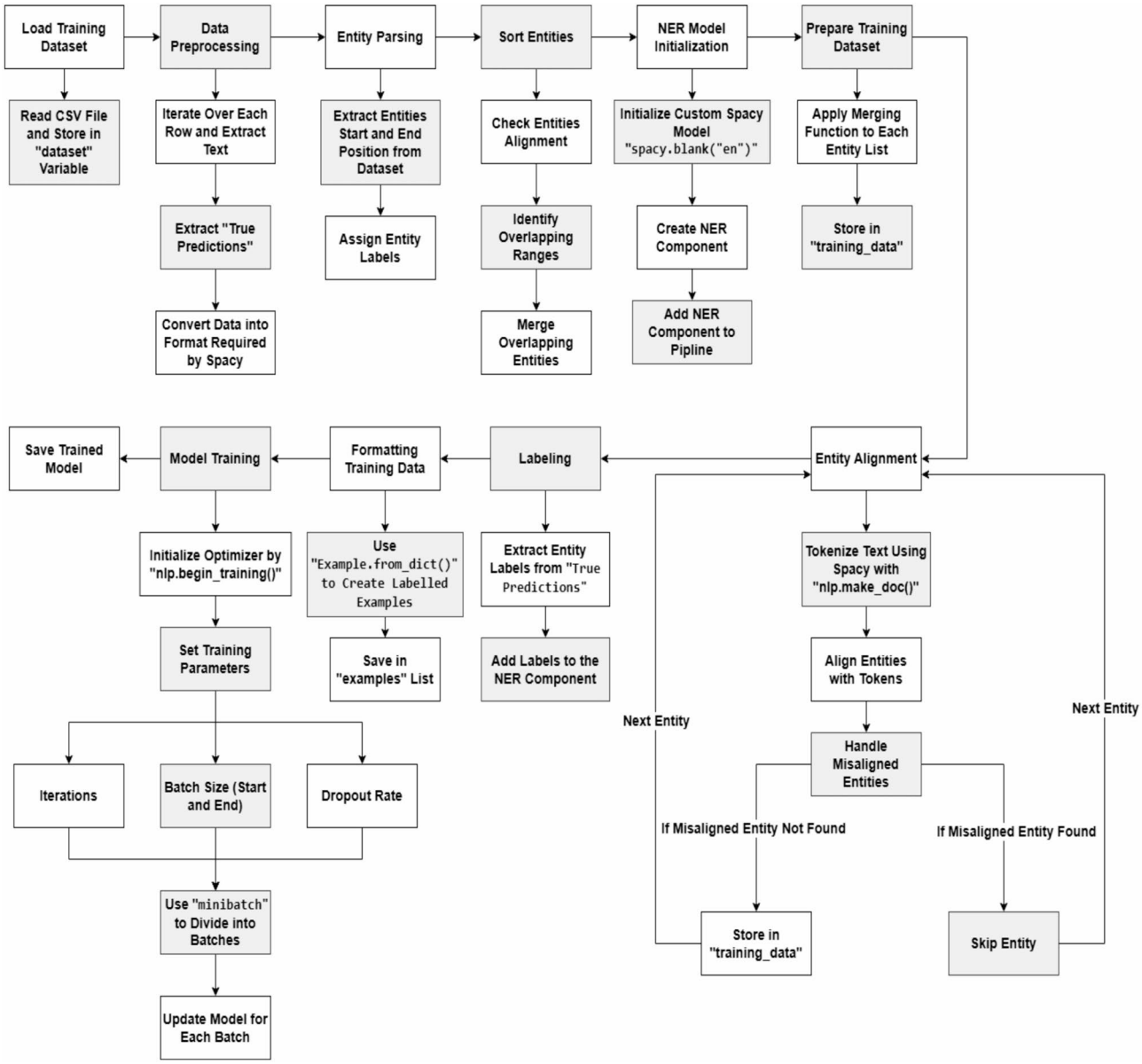

Fig. 5. Workflow for training the custom named entity recognition (NER) model.

32 examples per batch (compounding rate = 1.001). A dropout probability of 0.50 (*drop = 0.5*) was enforced to regularize the model and mitigate overfitting. For every mini-batch, model weights were updated through *nlp. update(batch*, *losses = losses*, *drop = 0.5*, *sgd = optimizer)* thereby minimizing cumulative loss across batches and refining the model's capacity to recognize and classify PII entities.

The core of the training process consists of iterations, where the model learns from the training data. On each iteration, the training examples are divided by minibatch into batches, and the model is updated with the help of these batches. The training loop can be mathematically described as follows:

$$\text{Loss}_{\text{total}} = \sum_{i=1}^{\text{iterations}} \sum_{b \in B_i} \text{Loss}\,(b) \tag{7}$$

.

Here, 'iterations' is the total number of training iterations, 'Bi' represents the batches in the 'i-th' iteration and Loss(b) is the loss computed for batch 'b'. The training loop aims to minimize this overall loss, which would improve the capability of the model to recognize PII.

Using *nlp.to_disk()*, the model saves to disk at the end of training iterations so that this trained model can be reused for new text samples to find and classify PII in later inference tasks.

Some key steps covered in this section for the process of training have to do with the preparation and formatting of training data, handling overlapping entities, creating a model, and running an iterative training

loop. Painstakingly combining overlapping entities and configuring a special spaCy model creates a strong PII detection model that keeps being refined after many iterations.

This is a wide training process, and the resulting model will be very reliable for PII detection. This model can detect objects it has encountered in training and generalize into finding patterns of text it has never seen.

Training environment

To ensure reproducibility of the above training regimen and to contextualize the reported performance, the model was developed and evaluated under the following training environment as mentioned in Table 2.

A complete requirements.txt and environment setup script are provided in the project repository to facilitate exact replication of this configuration (Section VII).

The model could serve as a dependable tool in handling sensitive data by efficiently identifying and classifying PII in various documents. Guaranteeing the accuracy and adaptability of the final model to various text settings is done by meticulous data preparation, model configuration, and iterative training. This planned course of training guarantees the model's proficiency in reliably detecting and classifying various types of PII. The trained NER model has considerable predictive reliability due to methodical data preparation, precise entity alignment, and iterative optimization with spaCy, serving as the foundation for further evaluation and anonymization processes. Following the successful training of the proposed model, thorough testing and evaluation are important to assess its generalization abilities. The upcoming subsection demonstrates the validation of model's performance using an independent testing dataset as well as testing on some real world available document, assessing its predictive efficacy through comprehensive quantitative indicators.

Executing code for desired outcomes

Here, the authors test the performance of the NER model using the real world audit report and invoices as well as testing dataset created. Testing includes several steps, such as loading the trained model, applying it to the testing data, extracting the predictions, and comparing them with the actual annotations to produce performance metrics. This evaluation will help us to understand the generalization capability of the model on new data for the identification of correct entities related to PII. Metrics such as accuracy, precision, recall, F1-score, and recall will show us how well this model works in segregating various types of PII. Then, it goes through each entry in the test dataset and predicts the entities that are found in that text. The success of the model is then measured by comparing these predictions with the actual labels or annotations. Key metrics are computed as a measure of the model's detection capability, and results are stored for future study. These steps are now deconstructed in detail, together with an explanation of the related flowchart presented in Fig. 6, showing the test cycle from loading the model to saving results. As shown in Fig. 6, the testing and evaluation workflow initiates by loading the trained model and the testing dataset containing annotated text samples. For each text entry, the model predicts the entities found within the text. A new column is added along side the original annotations to hold the predicted entities. The ability of the model to identify entities is then tested by comparing predicted annotations with actual ones. Based on the comparison, vital performance metrics like Accuracy, Precision, Recall, and F1 Score are calculated to understand how effective the model is. Finally, the results that comprise predictions and

Category	Parameter	Value
Hyper-parameter	Iterations (full corpus passes)	20
	Dropout	0.50
	Mini-batch start size	4
	Mini-batch end size	32
	Compounding rate	1.001
	Optimiser algorithm	spaCy Adam (decoupled weight-decay schedule)
	Learning-rate	0.001
	β_1 / β_2	0.9 / 0.999
	ε	1×10^{-8}
	Weight-decay (L2)	0.0
Model labels	Entity set	NAME, CREDIT_CARD, EMAIL, URL, PHONE, ADDRESS, COMPANY, SSN
Hardware	CPU	Intel i5-10300 H @ 2.5 GHz
	GPU	Nvidia GTX 1650 Ti (4 GB)
	RAM	8 GB DDR4
Software	Python	3.11.12
	spaCy	3.8.5
	Faker	28.0.0
	Pandas	2.2.2
	Scikit-learn	1.6.1
	Openpyxl	3.1.5
Runtime	Total training time	~7–8 h (CPU-only)

Table 2. Training environment of model.

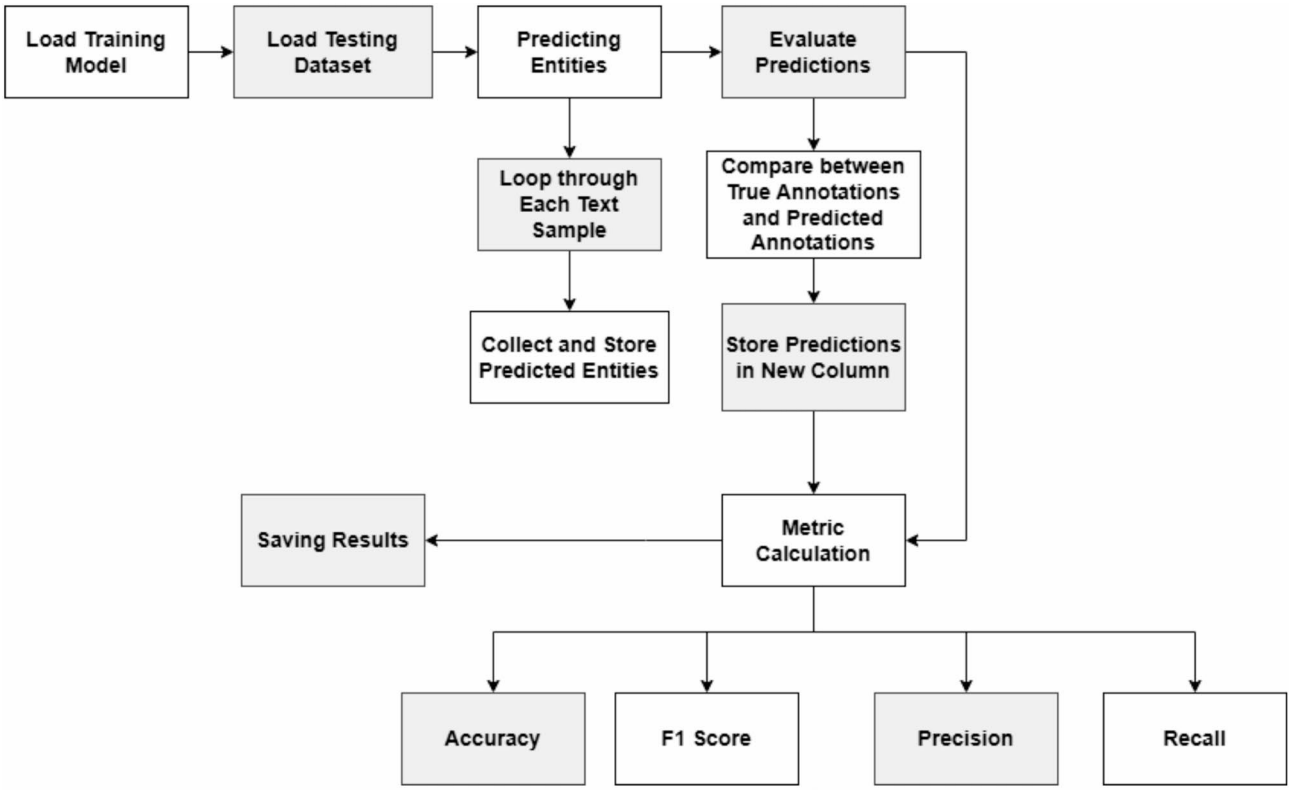

Fig. 6. Workflow for testing and evaluating the PII detection model.

metrics calculated are stored in a systematic manner so that they could be viewed and inspected again later. The systematic process ensures that the model is tested rigorously to determine whether it can identify and classify PII within unseen data.

Firstly, a trained NER model is loaded from the directory in which it has been saved. In this exercise, this was done by training the model on several types of PII in text. In addition, test dataset will be loaded containing the text data and the correct PII annotation. By doing this, the model and data are set for processing, and proper ground has been given at the start of the evaluation. Following the model and data loading, predictions are made for every text entry in the dataset. After processing each phrase and detecting any entities inside of it, the model outputs an entity type and start and end character indices. In this process, the entities 'P' for a given text 'X' is predicted using the trained model 'M':

$P(X) = M(X)$, where the set of expected entities, with their types and locations, is part of $P(X)$.

The predictions from the model are formatted to fit the original form of the testing dataset. That is the mapping of the detected entities to their corresponding types, "name," "email," and "phone," among others. Then, these are logged for every text entry in the dataset. In this way, the output will be in a format where one can see what items have been recognized by the model and will be able to compare these against actual annotations. The authors will also take the actual annotation in the test dataset against the prediction done by the model. Along with extracting the entities predicted by the model for every 'text' input, the actual entities were compared with those manually annotated. This comparison becomes necessary to see just how precise the model is in detecting the PII. Start by creating two sets: the model's true and predicted annotations. By comparing these sets, will give True Positives (TP). False Positives (FP). False Negatives (FN). These counts form the basis of computation for performance measures. The formula that might be used in this step to evaluate predictions can be mathematically expressed as follows:

True Positives = $|A \cap P|$, where False Positives = $|P - A|$ and False Negatives = $|A - P|$.

Here, 'A' is the actual set of annotations, and 'P' represents the predicted set of annotations. This score will depict the model's performance in identifying things like PII. Following the analysis of the predictions, the next step would be the calculation of performance metrics, which generally allow for a numerical evaluation of the model's effectiveness. The metrics are essential for understanding the performance of the model for various aspects of entity detection:

- Precision: The ratio of correctly identified entities among all the entities the model predicts. This gives us an idea of how many entities were found correctly.
- Recall: The metric tells you the percentage of real PII entities that the model correctly identified. It basically tells you about how many of the relevant entities the model captures.

Report name	True results	Predicted results
Synthetic audit report template	[(3482, 3494, 'name'), (5162, 5174, 'name'), (5178, 5202, 'email'), (2462, 2490, 'url'), (5206, 5220, 'phone'), (926, 969, 'address'), (2047, 2090, 'address'), (3744, 3787, 'address'), (5283, 5326, 'address'), (80, 94, 'company'), (440, 454, 'company'), (2008, 2022, 'company'), (5027, 5041, 'company'), (5386, 5400, 'company'), (1627, 1638, 'ssn'), (2820, 2831, 'ssn'), (3119, 3130, 'ssn')]	[(80, 94, 'company'), (440, 454, 'company'), (926, 969, 'address'), (1627, 1638, 'ssn'), (2008, 2022, 'company'), (2047, 2090, 'address'), (2462, 2490, 'url'), (2820, 2831, 'ssn'), (3119, 3130, 'ssn'), (3482, 3494, 'name'), (3744, 3787, 'address'), (5027, 5041, 'company'), (5162, 5174, 'name'), (5178, 5202, 'email'), (5206, 5220, 'phone'), (5283, 5326, 'address'), (5386, 5400, 'company')]
Synthetic vendor transaction confirmation template	[(5, 15, 'name'), (79, 138, 'credit_card'), (176, 190, 'phone'), (46, 60, 'company')]	[(5, 15, 'name'), (46, 60, 'company'), (79, 139, 'credit_card'), (176, 190, 'phone')]

Table 3. True results vs. predicted results for synthetic dataset—Part 01.

Report name	Name	Phone number	Email	Address	SSN	Credit card	Company name	URL
Synthetic audit report template	Andrew Yoder	+91 1,472,004,612	karensalazar@example.org	901 Austin Well Apt. 454 Cookport, WY 33,980	575-74-7674		Jackson-Guzman	http://www.moore-orozco.com/
Synthetic vendor transaction confirmation template	David Owen	+91 3,663,803,403				VISA 16 digit Melissa Dunn 4,618,265,793,875,936 12/26 CVC: 564	Reilly-Camacho	

Table 4. PII detected in synthetic dataset—Part 02.

Report name	True results	Predicted results
Synthetic audit report template	[(2396, 2445, 'address'), (18, 43, 'company'), (1812, 1837, 'company')]	[(18, 43, 'company'), (1812, 1837, 'company'), (2396, 2445, 'address')]
Synthetic risk assesment and compliance report template	[(3777, 3789, 'name'), (3699, 3760, 'credit_card'), (3175, 3192, 'email'), (3810, 3828, 'url'), (3196, 3210, 'phone'), (3255, 3309, 'address'), (96, 124, 'company'), (2840, 2868, 'company'), (3432, 3460, 'company'), (3365, 3376, 'ssn')]	[(96, 124, 'company'), (2840, 2868, 'company'), (3175, 3192, 'email'), (3196, 3210, 'phone'), (3255, 3309, 'address'), (3365, 3376, 'ssn'), (3432, 3460, 'company'), (3699, 3760, 'credit_card'), (3777, 3789, 'name'), (3810, 3828, 'url')]
Synthetic receipt confirmation template	[(160, 184, 'email'), (53, 95, 'address'), (15, 41, 'company')]	[(15, 41, 'company'), (53, 95, 'address'), (160, 184, 'email')]
Synthetic bank confirmation template	[(74, 93, 'url'), (8, 19, 'ssn')]	[(8, 19, 'ssn'), (74, 93, 'url')]
Synthetic ITR and tax return report template	[(60, 70, 'name'), (1177, 1187, 'name'), (872, 933, 'credit_card'), (1191, 1210, 'email'), (1307, 1326, 'url'), (1214, 1228, 'phone'), (134, 184, 'address'), (367, 376, 'company'), (591, 600, 'company'), (109, 120, 'ssn')]	[(60, 70, 'name'), (109, 120, 'ssn'), (134, 184, 'address'), (367, 376, 'company'), (591, 600, 'company'), (872, 934, 'credit_card'), (1177, 1187, 'name'), (1191, 1210, 'email'), (1214, 1228, 'phone')]

Table 5. True results vs. predicted results for synthetic dataset—Part 01.

- F1-score: This considers the number of false positives and false negatives, and it therefore gives a balanced measure of the performance of the model.
- Accuracy: The total percentage of correct forecasts, i.e., the sum of true positives and true negatives, out of all forecasts.

The values of all these metrics are shown in the Results and Discussion Section for review.

The last step entails writing results, performance measures, and forecasts onto an output file so that extensive examination and evaluation of model performance can be done, highlighting strong points and unveiling possible lines of improvement. An Excel file containing the result is maintained for ease of access and review.

Table 3: Part 01 and Table 4: Part 02 shows the first part of the results from the PII Detection Model. In the data, different entities are represented by separate columns: Name, Phone Number, Email, Address, SSN, Credit Card, Company Name, and URL. Predicted Results show where the places were estimated to be according to the model; True Results enumerate the actual places of PII elements in the text. Each row represents a different test case of different kinds of financial reports/templates. For instance, for the first kind, the model correctly identified the person, firm, and URL. Strong entity recognition accuracy is reflected by the values within the Predicted Results column being very similar to those within the True Results column. Specific values of PII identified by the model appear in the table below, which gives an overview of the PII in each text sample. Examples of such specific values are phone numbers and email addresses.

Further execution of test cases of the model is also depicted in Table 5: Part 01 and Table 6: Part 02. Similar to the previously discussed sections, it also carries real annotations and model predictions and has columns for different categories of PII. Certain rows, like 4th and 5th rows from the Table 6—Part 02 in this model, signify that the model can correctly and effectively detect complicated data as well, including credit card details and SSNs. Taken together, these tables show the robustness of the model in identifying different types of PII across

Report name	Name	Phone number	Email	Address	SSN	Credit card	Company name	URL
Synthetic audit report template				32,836 Anthony Park Suite 592 Pollardton, IL 96,104			Hogan, Smith and Galloway	
Synthetic risk assesment and compliance report template	Bryce Willis	+91 4,511,950,664	amy47@example.com	729 Sims Extensions Apt. 382 Lake Brianshire, MO 17,118	283-43-8647	VISA 16 digit Joseph Bentley 4,960,389,176,757,513 06/31 CVC: 456	Scott, Thompson and Schaefer	http://benson.net/
Synthetic receipt confirmation template			moorevincent@example.org	3792 Carlos Center Cooperchester, AL 36,727			Anderson, Hill and Coleman	
Synthetic bank confirmation template					130-67-8658			https://larson.com/
Synthetic ITR and tax return report template	Ryan Ortiz	+56 8,366,018,544	jason52@example.org	714 Kane Forks Apt. 893 Port Richardport, PW 24,345	299-55-3712	JCB 16 digit Brandon Mullins 3,536,895,160,092,771 04/34 CVC: 157	Tyler LLC	

Table 6. PII detected in synthetic dataset—Part 02.

Report name	True results	Predicted results
Bank of America audit report	[(0,21,'company'), (150, 187, 'company'), (15807, 15831, 'company'), (15838, 15858, 'company'), (16042, 16060, 'name'), (16062, 16074, 'name'), (16434, 16453, 'company'), (22407, 22431, 'company'), (22438, 22458, 'company'), (22639, 22657, 'name'), (22659, 22671, 'name'), (24267, 24291, 'company'), (24297, 24317, 'company'), (24553, 24571, 'name'), (24573, 24585, 'name'), (24587, 24599, 'name'), (24601, 24614, 'name'), (28767, 28791, 'company'), (28797, 28817, 'company'), (29052, 29070, 'name'), (29072, 29084, 'name'), (29086, 29098, 'name'), (29100, 29113, 'name'), (33056, 33077, 'company')]	[(0,21,'company'), (150, 187, 'company'), (15807, 15831, 'company'), (15838, 15858, 'company'), (16042, 16060, 'company'), (16062, 16074, 'name'), (16434, 16453, 'company'), (22407, 22431, 'company'), (22438, 22458, 'company'), (22639, 22657, 'name'), (22659, 22671, 'name'), (24267, 24291, 'company'), (24297, 24317, 'company'), (24553, 24571, 'name'), (24573, 24585, 'name'), (24587, 24599, 'name'), (24601, 24614, 'name'), (28767, 28791, 'company'), (28797, 28817, 'company'), (29052, 29070, 'name'), (29072, 29084, 'name'), (29086, 29098, 'name'), (29100, 29113, 'name'), (33056, 33077, 'name')]
Amazon vendor invoice	[(504, 510, 'company'), (720, 726, 'name'), (978, 1017, 'company'), (1019, 1086, 'address'), (1414, 1420, 'name'), (1422, 1461, 'company'), (1463, 1530, 'address'), (1817, 1836, 'url'), (225, 261, 'company'), (263, 300, 'company'), (603, 670, 'address')]	[(504, 510, 'company'), (720, 726, 'name'), (1019, 1086, 'address'), (1414, 1420, 'name'), (1422, 1461, 'company'), (1463, 1530, 'address'), (1817, 1836, 'url'), (225, 261, 'company'), (263, 300, 'company'), (603, 670, 'address')]
Musterkunde AG vendor invoice	[(0, 37, 'company'), (38, 51, 'company'), (52, 63, 'name'), (64, 79, 'address'), (80, 99, 'address'), (106, 122, 'name'), (130, 148, 'phone')]	[(0, 37, 'company'), (38, 51, 'company'), (64, 79, 'address'), (80, 99, 'address'), (106, 122, 'name'), (130, 148, 'phone')]

Table 7. True results vs. predicted results in real world documents—Part 01.

different scenarios. In fact, for most entities, the model's predictions match the actual outcomes, which shows that this model easily generalizes the training data to new examples.

Hence, this subsection presents the steps for carrying out testing of the PII detection model with respect to effectiveness through a test dataset. From the results, it can be seen that this model can effectively detect and extract PII from various text samples, with a good Accuracy of 89.4%, Precision of 94.7%, and Recall of 89.4%. The model is suited to any real-world application where sensitive information needs to be correctly detected, with robust performance metrics through a comprehensive study of the data. With a high F1 score and accuracy, the model displays the ability to strike that fragile balance between accurately detecting PII and preventing false positives.

For the study to validate the model's applicability beyond synthetic datasets, additional evaluation was conducted using different real-world financial documents, including a publicly available Bank of America audit report and two vendor invoices. Manual annotations was performed to establish ground truth for these documents. The model achieved an accuracy of 93% on these real-world datasets, successfully detecting a range of PII categories such as Person Names, Company Names, Addresses, Phone Numbers, and URL. The results show that the model performs well in real-life financial reports and can be used with a wide range of complex text forms and documents in real-time applications. Real world financial report testing is visualized in Table 7—Part 01 and Table 8: Part 02.

The last phase, focuses on the anonymization of identified Personally Identifiable Information (PII) to provide privacy protection. The following section delineates the anonymization method, highlighting the systematic substitution of sensitive data pieces with generic placeholders to maintain data security while preserving textual coherence.

Anonymization of PII data in texts

While handling PII, text anonymization becomes an important part for data protection since it protects sensitive personal information. The proposed model can make sure that any personally identifiable information (PII) in financial records is properly anonymized so that people's privacy is protected. In this exercise, generic placeholders

Report name	Name	Phone number	Email	Address	SSN	Credit card	Company name	URL
Bank of America audit report	Vinit K Jain; Murad D. Daruwalla; Kaku Nakhate; Viral Damania; Reserve Bank of India	–	–	–	–	–	BANK OF AMERICA, N.A. BANK OF AMERICA, N.A. (INDIA BRANCHES); Walker Chandiok & Co LLP; KKC & Associates LLP; Murad D. Daruwalla; Bank of America N.A.	
Amazon vendor invoice	Madhu B	–	–	Eurofins IT Solutions India Pvt Ltd, 1st Floor, Maruti Platinum, Lakshminarayana Pura, AECS Layout, BENGALURU, KARNATAKA, 560,037, IN; Varasiddhi Silk Exports, 75, 3rd Cross, Lalbagh Road, BENGALURU, KARNATAKA, 560,027, IN	–	–	Varasiddhi Silk Exports; Eurofins IT Solutions India Pvt Ltd. Amazon Seller Services Pvt. Ltd.; Amazon Retail India Pvt. Ltd.	amazon.in/business
Musterkunde AG vendor invoice	Stefanie Müller	+49 9371 9786-0	–	Musterstr. 23, 12,345 Musterstadt	–	–	CPB Software (Germany) GmbH, Musterkunde AG	–

Table 8. PII detected in real world documents—Part 02.

are used for the identified PII entities to ensure that the data is secure for further processing or dissemination. The anonymization process protects sensitive data, such as names, phone numbers, email addresses, and other identifiers, by preventing their accidental disclosure. The following section elaborates on the procedures that were applied to anonymize PII in the proposed dataset with textual integrity intact and full protection of privacy.

Figure 7 describes the anonymization procedure. The anonymization workflow begins by loading the dataset containing original texts and predicted PII results. The annotations, which indicate the positions of detected PII entities, are searched within each original text document. Along with the positional information, the type of each PII entity (such as Name, Email, Address) is gathered from the predicted results. The anonymization process replaces the identified sensitive information with appropriate placeholders like "[NAME REDACTED]" or "[ADDRESS REDACTED]" within the text by using the type and location. When the replacement is completed, the anonymized versions of the documents are saved into a new column, which ensures a structured comparison between the original and anonymized data. This stepwise method guarantees accurate anonymization while also preserving the document's overall format and meaning. Next the results of data produced by model predictions have to be loaded. It then reads an Excel file, 'Results.xlsx,' containing the original texts and the PII predicted. These are preprocessed to prepare them for anonymization. For consistency with the naming convention used, the column 'True Predictions' has been renamed 'True Results.' This step is important because it ensures the data are in a format that is in line with the processing the data are to undergo and lays the foundation for real processing. Next comes the creation of a new section in the Excel workbook titled "Audit Reports." The purpose is mainly to keep the original texts in the dataset before any anonymization is affected. Columns that are part of the result DataFrame are dropped and copied into a new DataFrame, which is then renamed 'Text.' For the purpose of this section, the original data is referred to, allowing for a clear comparison between text in both its anonymized and unaltered state. This original data should be kept for audit and verification purposes. After that, the "Predicted Results" section needs to be created in the Excel workbook. This section is aimed at fully explaining the entities that this model predicts. Once the data is changed, additional columns are added, such as "True Results," "Predicted Results," "Name," "Phone Number," "Email," "Address," "SSN," "Credit Card," "Company Name," and "URL". Then, these columns are renamed according to the required format. On the other hand, accurate and complete documentation of the predictions forms the very foundation through which the model's output is first validated to ensure the anonymization procedure targets identified PII entities appropriately.

This block is the heart of the anonymization procedure. A function named anonymize_text, defined in the code, takes the original text and the model's prediction as input. It uses a dictionary that maps PII entity types into redacted placeholders, for instance, [NAME REDACTED], [EMAIL REDACTED], and so on. The function iterates over the expected entities individually; for each occurrence in the text, it replaces the correct placeholder. In order not to change indices, the replacement is carefully conducted from the end of the text toward the beginning, making sure that each found entity is anonymized correctly. Once the anonymization process is complete, this anonymized data should be placed into a new DataFrame. This DataFrame includes both the original and anonymized sentences for a direct comparison. The anonymize_text will be run on each row within the text to ensure all PII is properly redacted. One of the key outcomes of the anonymization process is the output DataFrame; this includes a secure version of the data that will not violate any of the data privacy legislation. Saving the results into an Excel file is an end to anonymization. The anonymized texts are written onto a new worksheet called "Anonymized Data." This worksheet will be created using the code in the Excel workbook. That is to say that a good format of the Excel file can serve to present the data in a very easy-to-use manner. The final workbook, Results.xlsx, includes tabs for "Audit Reports," "Predicted Results," and "Anonymized Data." This full document will make the anonymization process transparent and traceable with a detailed summary of the original, forecasted, and anonymized data.

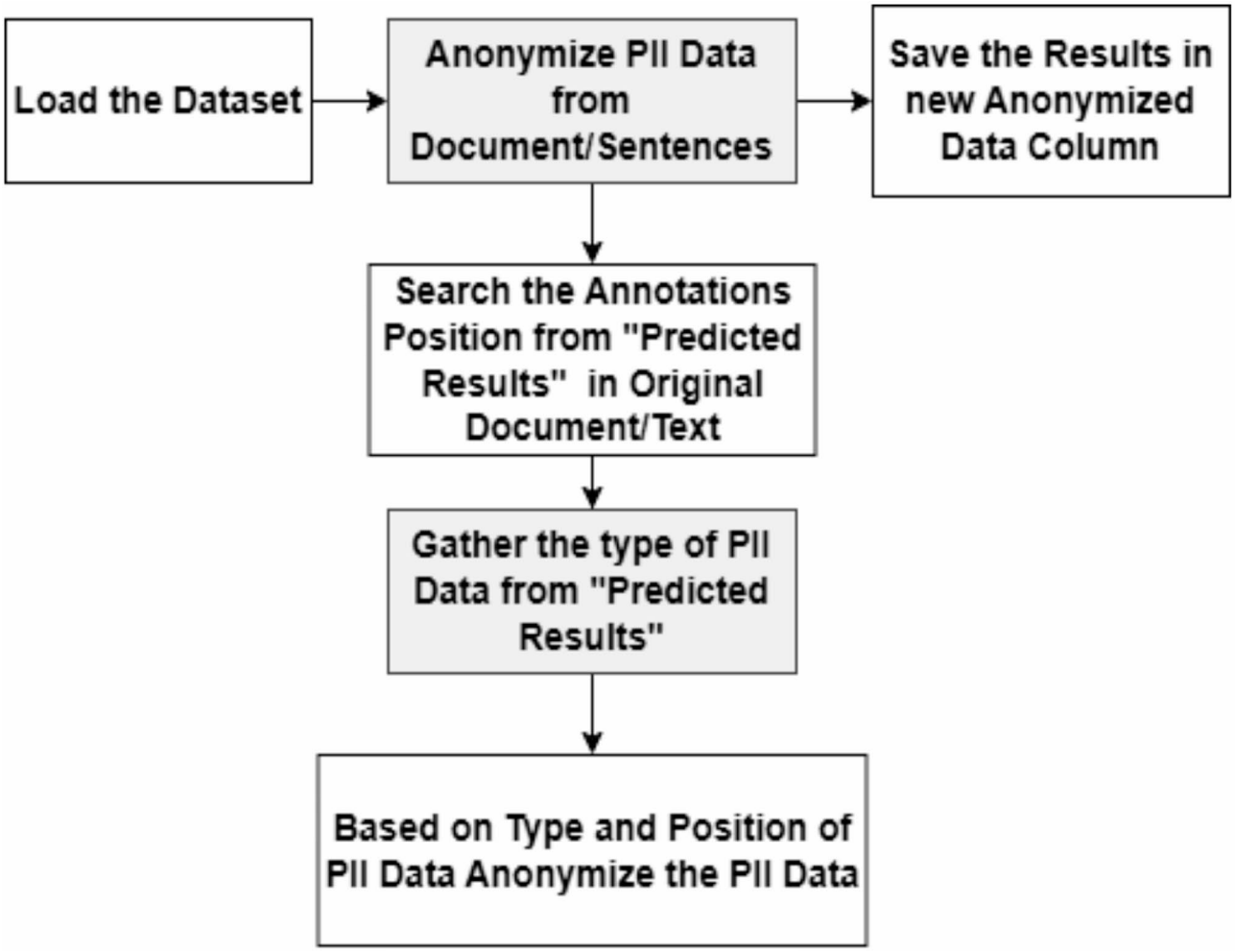

Fig. 7. Anonymization workflow for sensitive data.

The results of such an anonymization process are depicted in the following Figures, where all the placeholders have been used instead of the original PII to anonymize it successfully and make any illegal access to personal information invalid.

Figure 8 shows the anonymization of an synthetic audit report. The left column is the text with the Company Name, along with specific Addresses and Social Security Numbers, intact. The anonymized text used in this phase of the project is on the right-hand side. Here, all PII has been replaced with very general placeholders such as "[COMPANY NAME REDACTED]" or "[ADDRESS REDACTED]" or "[SSN REDACTED]". This change demonstrates how the system recognizes and removes sensitive data, securing it for analysis or dissemination while maintaining privacy.

Figures 9 and 10 show some synthetic anonymized text examples on transaction data and client comments. The original text includes addresses, personal names, and credit card numbers. The anonymized version shows the system's ability to handle different types of sensitive information in diverse settings effectively by replacing these facts with such phrases as "[CREDIT CARD REDACTED]" and "[NAME REDACTED]." This guarantees confidentiality and compliance with the law on data privacy in that it would not be possible to trace any personal information back to particular individuals. These numbers outline how the technology can anonymize documents by itself, while any personally identifiable information will be kept safe, and the basic structure and sense of the document will be preserved.

To further re-validate the robustness of the proposed anonymization approach on real-world financial documents, two representative examples were selected and processed: an independent auditor's report of Bank of America N.A. (India Branches) and an invoice issued by Amazon Seller Services Pvt. Ltd. (ASSPL) and Amazon Retail India Pvt. Ltd. (ARIPL). As described in Fig. 11, the anonymization flow successfully identifies and masks sensitive information such as company names, client references, and URLs while preserving the overall semantic structure and format of the documents. Notably, all instances of "Bank of America" and "Amazon Seller Services Pvt. Ltd." were systematically redacted to [COMPANY NAME REDACTED], and web links were replaced with [URL REDACTED] wherever applicable.

This displays the capacity of the proposed model to recognize entities embedded in both formal audit narratives and transactional invoice formats, emphasizing its versatility across different types of financial text.

Original Text	Anonymized Text
We have conducted a thorough review of the tax compliance practices followed by Jackson-Guzman for the fiscal year ending March 31, 2023. Our examination included a detailed analysis of corporate tax returns, GST filings, and withholding tax submissions across all divisions. The review focused on ensuring compliance with the latest amendments in tax laws and regulations. Corporate Tax Overview The corporate tax computation for Jackson-Guzman was cross-verified against the financial statements audited by our internal team. The tax liability was calculated considering various deductions under section 80C, 80D, and other relevant sections of the Income Tax Act. The total taxable income stood at INR 500 Crores, with an effective tax rate of 25%. Key points include: Depreciation Deductions: Claimed as per the Income Tax Act, aligned with the rates prescribed under Schedule II. The assets located at 901 Austin Well Apt. 454 Cookport, WY 33980 were correctly depreciated using the Written Down Value (WDV) method. The details of high-value assets have been corroborated with the asset register maintained at the corporate office. Tax Credits: The company has utilized carry-forward losses from previous financial years to offset the current tax liability, reducing the net payable tax. The adjusted tax liability has been duly filed with the tax authorities. Deductions: The deductions for contributions to the Employee Provident Fund (EPF) and Gratuity are in compliance with sections 80C and 80D. However, we noted a delay in the deposit of EPF contributions for some employees whose SSNs 575-74-7674 end with '4567'. This delay has been flagged, and a provision for potential interest and penalties has been recommended.	We have conducted a thorough review of the tax compliance practices followed by [COMPANY NAME REDACTED] for the fiscal year ending March 31, 2023. Our examination included a detailed analysis of corporate tax returns, GST filings, and withholding tax submissions across all divisions. The review focused on ensuring compliance with the latest amendments in tax laws and regulations. Corporate Tax Overview The corporate tax computation for [COMPANY NAME REDACTED] was cross-verified against the financial statements audited by our internal team. The tax liability was calculated considering various deductions under section 80C, 80D, and other relevant sections of the Income Tax Act. The total taxable income stood at INR 500 Crores, with an effective tax rate of 25%. Key points include: Depreciation Deductions: Claimed as per the Income Tax Act, aligned with the rates prescribed under Schedule II. The assets located at [ADDRESS REDACTED] were correctly depreciated using the Written Down Value (WDV) method. The details of high-value assets have been corroborated with the asset register maintained at the corporate office. Tax Credits: The company has utilized carry-forward losses from previous financial years to offset the current tax liability, reducing the net payable tax. The adjusted tax liability has been duly filed with the tax authorities. Deductions: The deductions for contributions to the Employee Provident Fund (EPF) and Gratuity are in compliance with sections 80C and 80D. However, we noted a delay in the deposit of EPF contributions for some employees whose SSNs [SSN REDACTED] end with '4567'. This delay has been flagged, and a provision for potential interest and penalties has been recommended.

Fig. 8. Original vs. anonymized text—synthetic audit reports.

During the audit of the financial statements for Miller Ltd, it was observed that Mr Jenny Gilbert, the Chief Financial Officer, approved the purchase of assets worth $500,000 on Unit 3386 Box 5825 DPO AE 43903 The payment was processed through the credit card ending in Maestro Bradley Saunders 639052387128 10/32 CVV: 331 Mr Jenny Gilbert's Social Security Number (SSN) is 315-68-8485 For any clarifications, please reach out to Mr Jenny Gilbert at yespinoza@exampleorg or contact him directly at +91 8840920617 Further details can be accessed at https://cannonbiz/	During the audit of the financial statements for [COMPANY NAME REDACTED], it was observed that Mr [NAME REDACTED], the Chief Financial Officer, approved the purchase of assets worth $500,000 on [ADDRESS REDACTED] The payment was processed through the credit card ending in [CREDIT CARD REDACTED]Mr [NAME REDACTED]'s Social Security Number (SSN) is [SSN REDACTED] For any clarifications, please reach out to Mr [NAME REDACTED] at [EMAIL REDACTED] or contact him directly at [PHONE NUMBEI REDACTED] Further details can be accessed at https://cannonbiz/
Customer Kathy Hutchinson from Williams, Baker and Hunter, located at 75395 Anthony Pines South April, TX 37507, has submitted feedback regarding the recent transaction involving their credit card VISA 13 digit Monique James 4926423785049 11/24 CVC: 388 They can be reached at heathersmith@exampleorg or +211 9612363682 for further discussions The customer's Social Security Number (SSN) on file is 899-24-4737 The feedback was originally submitted through our website at http://wilkinsoncom/.	Customer [NAME REDACTED] from [COMPANY NAME REDACTED], located at [ADDRESS REDACTED], has submitted feedback regarding the recent transaction involving their credit card [CREDIT CARD REDACTED]They can be reached at [EMAIL REDACTED] or [PHONE NUMBER REDACTED] for further discussions The customer's Social Security Number (SSN) on file is [SSN REDACTED] The feedback was originally submitted through our website at [URL REDACTED].
This is a confirmation that the payment for the invoice number INV-409876535422 from Maddox Group has been successfully processed. The payment was made using the credit card ending in Discover Joe Villegas 6011001713881318 02/30 CVC: 631 by Andrea Ruiz. The billing address on file is 897 Flores Burgs Suite 750 North Rachelbury, FM 34563. The Social Security Number (SSN) for Andrea Ruiz is 577-02-7219. Should you have any inquiries, you may contact Andrea Ruiz via email at reyesdeborah@example.com or call +91 6289095696. For more information, visit http://hawkins-wilcox.info/	This is a confirmation that the payment for the invoice number INV-409876535422 from [COMPANY NAME REDACTED] has been successfully processed. The payment was made using the credit card ending in [CREDIT CARD REDACTED]by [NAME REDACTED]. The billing address on file is [ADDRESS REDACTED]. The Social Security Number (SSN) for [NAME REDACTED] is [SSN REDACTED]. Should you have any inquiries, you may contact [NAME REDACTED] via email at [EMAIL REDACTED] or call [PHONE NUMBER REDACTED]. For more information, visit [URL REDACTED]

Fig. 9. Original vs. anonymized text—synthetic transactional and customer feedback.

The redactions are comprehensive yet context-aware, ensuring that no accidental disclosures of confidential affiliations occur while maintaining document readability for audit or compliance purposes.

Along with this, another evaluation was performed on a detailed vendor invoice issued by CPB Software (Germany) GmbH, as shown in Fig. 12. The document contained Personally Identifiable Information (PII), including individual names, full addresses, and phone numbers. As described the anonymization framework effectively had masked all such sensitive fields by replacing names with [NAME REDACTED], addresses with [ADDRESS REDACTED], and phone numbers with [PHONE NUMBER REDACTED], while keeping transaction-critical details such as VAT numbers, invoice numbers, and monetary amounts intact. This type of selective anonymization as displayed make sures that privacy is maintained, the operational and financial meaning of the document remains unaffected. The evaluation highlights the model's ability to differentiate between PII and necessary financial data even in invoices that blend personal identifiers with structured billing information. The presented results in Figs. 11 and 12 confirm the proposed anonymization technique is not only effective on synthetic datasets but also generalizes well to complex real-world documents, demonstrating high practical applicability in audit, finance, and compliance domains. The anonymization procedure, as discussed in this section, for the most part, protects sensitive data while ensuring it does not disturb the structural integrity of the content, but it does prevent unauthorized disclosure by replacing the detected PII elements with generic placeholders. The result of this process is an detailed anonymized dataset after going through many steps, such as data preparation, anonymization itself, and documentation. This ensures the protection of data privacy policies and provides a secure framework needed to handle private information within financial documents. To encapsulate, the proposed anonymization workflow effectively replaces identified PII with contextually

We regret to inform you that a security breach was detected on Rodriguez-Vasquez's systems, which may have exposed your personal information, including your name (Ricardo Perez), email (crawfordchristian@example.net), phone number (+91 6034497453), and Social Security Number (SSN) (564-69-8794). The breach was traced back to unauthorized access from IP address 192.0.45. If you notice any suspicious activity on your credit card ending in VISA 16 digit Manuel Hernandez 4098912674666855 01/34 CVC: 631 , please contact us immediately. You can also check for updates on our security measures at http://sweeney.com/. The compromised data was stored at our facility located at 0851 Eileen Spring Suite 115 Lake Shannonport, MI 46578.	We regret to inform you that a security breach was detected on [COMPANY NAME REDACTED]'s systems, which may have exposed your personal information, including your name ([NAME REDACTED]), email ([EMAIL REDACTED]), phone number ([PHONE NUMBER REDACTED]), and Social Security Number (SSN) ([SSN REDACTED]). The breach was traced back to unauthorized access from IP address 192.0.45. If you notice any suspicious activity on your credit card ending in [CREDIT CARD REDACTED], please contact us immediately. You can also check for updates on our security measures at [URL REDACTED]. The compromised data was stored at our facility located at [ADDRESS REDACTED].
Dear Steven Patton, thank you for creating a new account with Gordon, Ford and Cunningham Your registered email is justin18@exampleorg, your contact number is +68 7123796774, and your Social Security Number (SSN) is 222-43-0194. The account was set up using the billing address USNS Orr FPO AA 39270, and the primary credit card linked to the account ends in VISA 16 digit Denise Young 4919315863789503 09/24 CVC: 164 . Please visit http://white.org/ to verify your account and update any personal details. If you need assistance, contact our support team	Dear [NAME REDACTED], thank you for creating a new account with [COMPANY NAME REDACTED] Your registered email is [EMAIL REDACTED], your contact number is [PHONE NUMBER REDACTED], and your Social Security Number (SSN) is [SSN REDACTED]. The account was set up using the billing address [ADDRESS REDACTED], and the primary credit card linked to the account ends in [CREDIT CARD REDACTED]. Please visit [URL REDACTED] to verify your account and update any personal details. If you need assistance, contact our support team
This Service Contract between Thompson, Gallagher and Richmond and Richard Diaz was entered at 068 Sherry Parkway Jacquelineburgh, FL 45032. The contract stipulates that all payments will be processed through the credit card provided by Richard Diaz, ending in Discover Michael Stout 6560045837280845 03/31	This Service Contract between [COMPANY NAME REDACTED] and [NAME REDACTED] was entered at [ADDRESS REDACTED]. The contract stipulates that all payments will be processed through the credit card provided by [NAME REDACTED], ending in [CREDIT CARD REDACTED]. [NAME REDACTED]'s Social Security Number (SSN) is [SSN REDACTED]. For further reference, correspondence will be sent to [EMAIL REDACTED], and all communications will be conducted via [PHONE NUMBER REDACTED]. The full contract details are available online at [URL REDACTED].

Fig. 10. Original vs. anonymized text—synthetic transactional and customer feedback.

Original Text	Anonymized Text
BANK OF AMERICA, N.A. (INDIA BRANCHES) (Incorporated in U.S.A. With Limited Liability) 1 Independent Auditor's Report To the Local Management Team of Bank of America N.A. (India Branches) Report on the Audit of the Financial Statements Opinion 1. WehaveauditedtheaccompanyingfinancialstatementsofBankofAmericaN.A(IndiaBranches)('theBank'),whichcomprisetheBalanceSheetas at 31 March 2023, the Profit and Loss Account, the Cash Flow Statement for the year then ended, and	[COMPANY NAME REDACTED] (INDIA BRANCHES) (Incorporated in U.S.A. With Limited Liability) 1 Independent Auditor's Report To the Local Management Team of [COMPANY NAME REDACTED] Report on the Audit of the Financial Statements Opinion 1. Wehaveauditedtheaccompanyingfinancialstatementsof[COMPANY NAME REDACTED]('theBank'),whichcomprisetheBalanceSheetas at 31 March 2023, the Profit and Loss Account, the Cash Flow Statement for the year then ended, and
Tax Invoice/Bill of Supply/Cash Memo (Original for Recipient) *ASSPL-Amazon Seller Services Pvt. Ltd., ARIPL-Amazon Retail India Pvt. Ltd. (only where Amazon Retail India Pvt. Ltd. fulfillment center is co-located) Customers desirous of availing input GST credit are requested to create a Business account and purchase on Amazon.in/business from Business-eligible offers Please note that this invoice is not a demand for payment Page 1 of 2	Tax Invoice/Bill of Supply/Cash Memo (Original for Recipient) *ASSPL-[COMPANY NAME REDACTED], ARIPL-[COMPANY NAME REDACTED] (only where [COMPANY NAME REDACTED] fulfillment center is co-located) Customers desirous of availing input GST credit are requested to create a Business account and purchase on [URL REDACTED] from Business eligible offers Please note that this invoice is not a demand for payment Page 1 of 2

Fig. 11. Original vs. anonymized text—audit report—Bank of America and invoice—Amazon Vendor.

CPB Software (Germany) GmbH - Im Bruch 3 - 63897 Miltenberg/Main Musterkunde AG Mr. John Doe Musterstr. 23 12345 Musterstadt Name: Stefanie Müller Phone: +49 9371 9786-0 Invoice WMACCESS Internet VAT No. DE199378386 Invoice No 123100401 Amount -without VAT- quantity 130,00 € 1 10,00 €	[COMPANY NAME REDACTED] - Im Bruch 3 - 63897 Miltenberg/Main [COMPANY NAME REDACTED] Mr. John Doe [ADDRESS REDACTED] [ADDRESS REDACTED] Name: [NAME REDACTED] Phone: [PHONE NUMBER REDACTED] Invoice WMACCESS Internet VAT No. DE199378386 Invoice No Customer No Invoice Period Date 123100401 12345 01.02.2024 - 29.02.2024 1. März 2024 Amount Service Description quantity Total Amount -without VAT- Basic Fee wmView 130,00 EUR 1 130,00 EUR Basis fee for additional user accounts 10,00 EUR 0 0,00 EUR Basic Fee wmPos 50,00 EUR 0 0,00 EUR Basic Fee wmGuide 1.000,00 EUR 0 0,00 EUR

Fig. 12. Original vs. anonymized text—invoice—vendor.

appropriate placeholders, maintaining both data privacy and textual coherence. The resulting anonymized documents retain their operational utility while significantly enhancing security and compliance, confirming the practical efficacy of the developed anonymization framework. After developing and testing the proposed PII detection and anonymization model, the authors now address the principal problems faced and the tactics employed to surmount them.

Mitigating challenges in PII detection and anonymization

The main objective of this work is to develop a comprehensive solution that can address various constraints found in state-of-the-art methods for detecting and anonymizing PII. Indeed, the proposed approaches have targeted problems like preserving semantic consistency, unstructured data processing, precision and recall during processing, reducing complex computation, and reducing demands for manual labeling while considering the challenges related to quasi-identifier handling. Table 9 highlights how the proposed technique successfully resolved each of these issues and points to improvements over prior approaches. The authors developed a system that could guarantee the practical usage of anonymized texts, improving the precision of PII identification through the embodiment of state-of-the-art NLP approaches, machine learning, and automated data production. The comprehensive approach demonstrates resiliency and adaptability in these scenarios:

Overall, by systematically addressing critical limitations of existing methodologies, such as precision-recall balance, semantic integrity, handling unstructured data, computational complexity, and reliance on manual annotation, the proposed hybrid approach significantly advances the state-of-the-art in PII detection and anonymization. The targeted improvements enable more reliable, secure, and operationally feasible management of sensitive data in financial context. Having developed and tested a comprehensive approach for identifying and anonymizing Personally Identifiable Information (PII), the proposed study now goes to a detailed analysis of the proposed model's performance. The following Results and Discussion section evaluates the efficacy of the methodology, offering thorough insights into the model's performance via several quantitative evaluations and visual representations.

Results and discussion

In this work, the performance of the model is tested against a suite of indicators that identify and anonymize Personally Identifiable Information. In order to get full insight into the pros and cons of this model, one should make use of precision-recall curves, ROC curves, confusion matrices, and overall performance metrics together. These analyses provide information not only on how well the model functions but also serve as some sort of action plan for further work in the future.

The precision-recall curve shown in Fig. 13 is fundamental for the effectiveness measure of the model. This is particularly relevant in scenarios with unbalanced data. Recall refers to the ability of the model to identify all the relevant appearances of PII, and precision displays the ratio of true positives among the predicted positives. In this chart, the line for each class draws the trade-off between recall and precision.

Interpretation: The closer the curve to the top-right corner, the better the model has the capability to give high recall and precision. In this picture, for example, classes like "email" and "phone" are nearly perfectly balanced to show the model's capability to detect those things accurately without compromising recall.

Another measure used to check the performance of the model is the ROC curve, presented in Fig. 14. It plots a graph of the true positive rate (sensitivity) versus the false positive rate (1-specificity) for different threshold settings. This measure can summarize the performance of the model into a single number with the use of Area Under the Curve (AUC).

Interpretation: An AUC of 0.5 reflects a performance no different from random guesses, while an AUC of 1 represents the high performance of a model. In fact, most of those classes, such as "address," "company," and "ssn," have very high AUCs close to 1 in this ROC, which means that this model has a high ability to distinguish the positive class from the negative one. The "url" class has a relatively lower AUC; hence, there is still room for improvement.

Figure 15 displays a confusion matrix, which provides comprehensive information about the performance of the model by showing the percentage share of correct and wrong class predictions. The diagonal element shows the correct prediction, and the off-diagonal elements are wrong classifications.

Limitation	How it was overcome
Balancing precision and recall	Proposed work shows a balance between these two entities of Precision and Recall, achieved by hybridizing between NLP and machine learning techniques. Iterative training and fine-tuning led to the optimal model performance through which both false positives and false negatives were minimized, hence catching the sensitive information with limited anonymization of non-sensitive data. Precision was 94.7%, while Recall was 89.4%.
Semantic consistency	Proposed work executed a strategic anonymization process wherein sensitive information was replaced with placeholders with extreme caution, maintaining the overall context and meaning of the document. Using templates and keeping the structure of the original sentences intact made sure that the anonymized texts were coherent and useful for further analysis, maintaining their semantic integrity.
Handling unstructured data	Proposed work is based on the use of current state-of-the-art NLP, complemented by custom spaCy NER models, which were trained on broad formats and structures of training data. This will represent the model's ability to pre-process and extract PII from unstructured documents such as e-mails and free-form documents, which provide better applicability in practical situations.
Computational complexity	The efficient training loop with optimized parameters was used to address the issue of computational complexity and to allow flexibility in terms of batch size. The authors have trained the model using spaCy, which offers a lightweight and fast implementation suitable for real-time and large-scale applications, hence providing scalability without sacrificing accuracy.
Handling quasi-identifiers	Proposed work consideres explicit PII and quasi-identifiers because they could lead to re-identification by proposing a comprehensive annotation scheme. The authors conduct comprehensive entity recognition followed by anonymization processes in the proposed approach to mask these indirect identifiers, which reduces the chances of disclosure of sensitive data through indirect means.
Manual annotation dependence	Proposed work minimizes manual annotation by generating a synthetic dataset using the Faker library, from which diversified PII examples can be automatically created. This way, not only is labor intensity reduced, but the quality of the training set is also very high because the model can easily learn in a great number of scenarios that otherwise require extensive manual input.

Table 9. Overcoming identified limitations in PII detection.

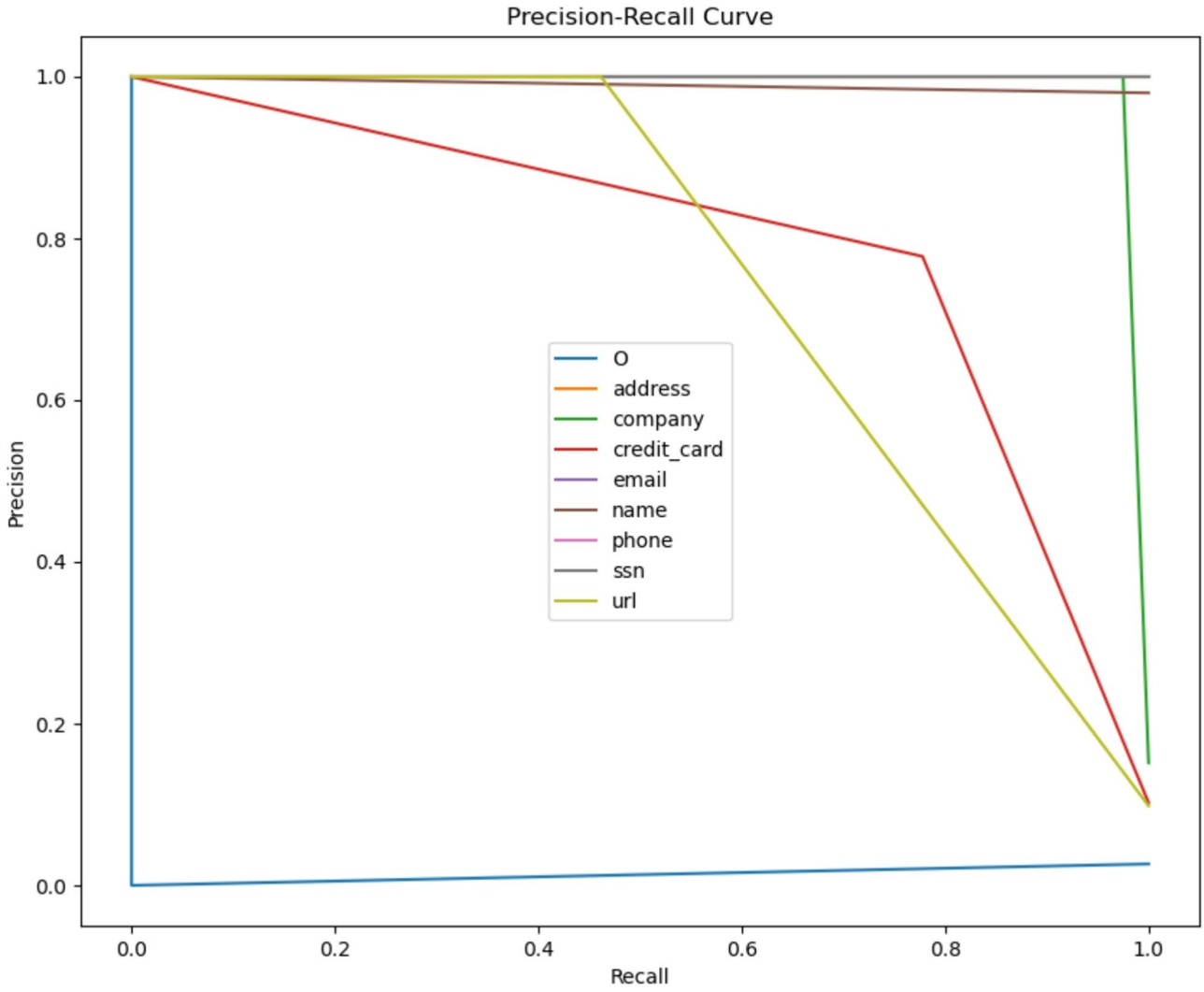

Fig. 13. Precision-recall curve.

Interpretation: From this heat map, it can be observed that most of the predictions are very close to the true labels except for "email," "name," and "phone." However, misclassifications in some classes, such as "credit" and "URL," are greater than in others. This chart helps to identify concrete areas lowering false positives for certain PII types in which the model's predictions could be improved.

Going through the confusion matrix (Fig. 15) shows certain categories where the proposed model struggled with false positives and negatives. The 'URL' category as shown in diagram had experienced a large number of false negatives, with 14 entries incorrectly labeled as not PII or 'O'. It suggests that URLs are a significant detection issue, most probably because of their diverse structural formats and similarity to harmless numeric or text patterns. Similarly, the 'credit_card' category experienced false negatives, six entries incorrectly labeled as not PII, pointing to the difficulty of accurately detecting credit cards within complex financial scenarios.

On the contrary, 'company,' 'email,' 'name,' 'phone,' and 'ssn' categories displayed outstanding detection, with minimal false positives or negatives, showing strong contextual and pattern recognition capabilities in these categories. These observations suggest where models can be improved, most critically through the development of better URL detection through sophisticated feature engineering or domain-based pattern analysis.

Figure 16 presents the four main performance metrics of the model in a bar graph form: accuracy, recall, F1 score, and precision. Each one of these statistics represents how well the model detects and anonymizes personally identifiable information.

- Precision (0.947): The precision score of the graph is pretty close to 0.95; that is, the model predicts the positive class with a very good amount of correctness. That can say that the chances of false positives are reduced because it detects PII correctly about 95% of the time.
- Recall (0.894): The recall score is around 0.89, putting it slightly lower than the precision one, meaning that actual instances of PII in the sample have been correctly retrieved by the model. In other words, the model normally catches PII but sometimes may let it slip through to create some false negatives.

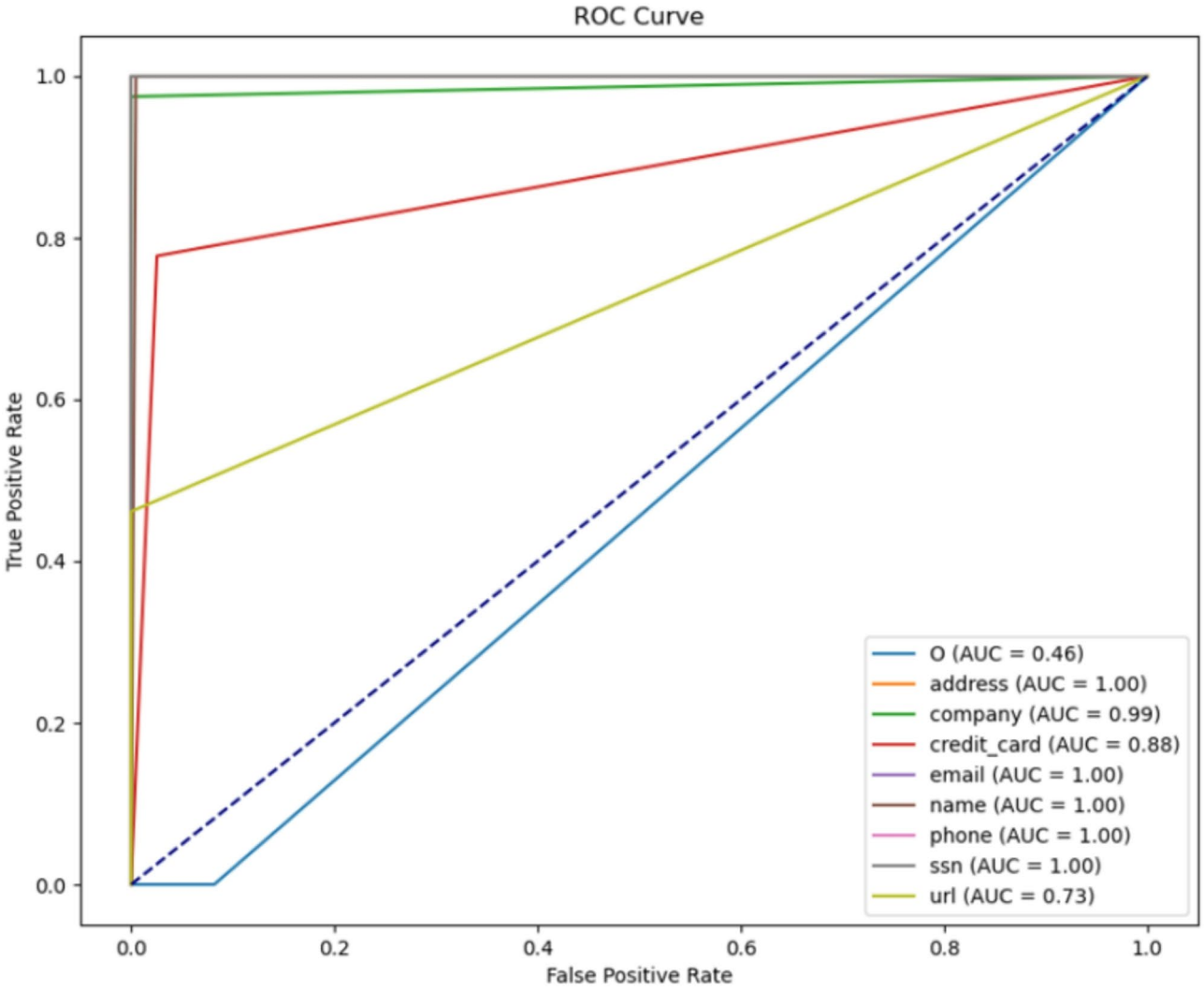

Fig. 14. Receiver operating characteristic curve.

- F1 Score (0.911): The F1 score is the balance between recall and precision at about 0.91. From this balancing score, in the case of the correct detection of personally identifiable information with the reduction of false positives, there is perceived steadiness in maintaining the balance. A high F1 score was the perfect overall measure of the model's efficacy.
- Accuracy (0.894): The model correctly predicted 0.89 of the total predictions, with either True Negatives or True Positives. It is evident that the model with this kind of accuracy can be trusted to make accurate general predictions for most PII kinds.

The proposed model was evaluated on both synthetically generated datasets and real-world financial documents. On synthetic datasets, the model achieved a precision of 94.7%, recall of 89.4%, and an F1 score of 91.1%. Testing on real-world documents, including a corporate audit report and vendor invoices, demonstrated consistent performance with approximately 93% accuracy. This dual evaluation strategy validates the model's capability to generalize effectively across different document types and text complexities.

Put together, these numbers should provide a basis for claiming that the model performs well in personally identifiable information recognition and protection, thus fitting well in practical use. The broader emphasis on model interpretability in sensitive domains[25] also supports the visualization and analysis approach adopted in this work.

Let us delve into Table 10, which compares the proposed hybrid ML-NLP based PII model against other widely used PII detection and anonymization techniques. The table covers essential features: Accuracy, Recall, F1-score, Precision, and particular benefits and drawbacks of each approach.

The efficiency of the proposed machine learning PII model in detecting different types of PII in unstructured data is validated by its high precision of 94.7% and a strong recall of 89.4%. In comparison, traditional approaches such as regex and rule-based systems often fail when encountering complex or varied unstructured data, leading to notably lower precision and recall values. While dictionary-based methods and certain machine learning approaches like SVMs and Random Forests demonstrate moderate effectiveness, they either suffer from limited

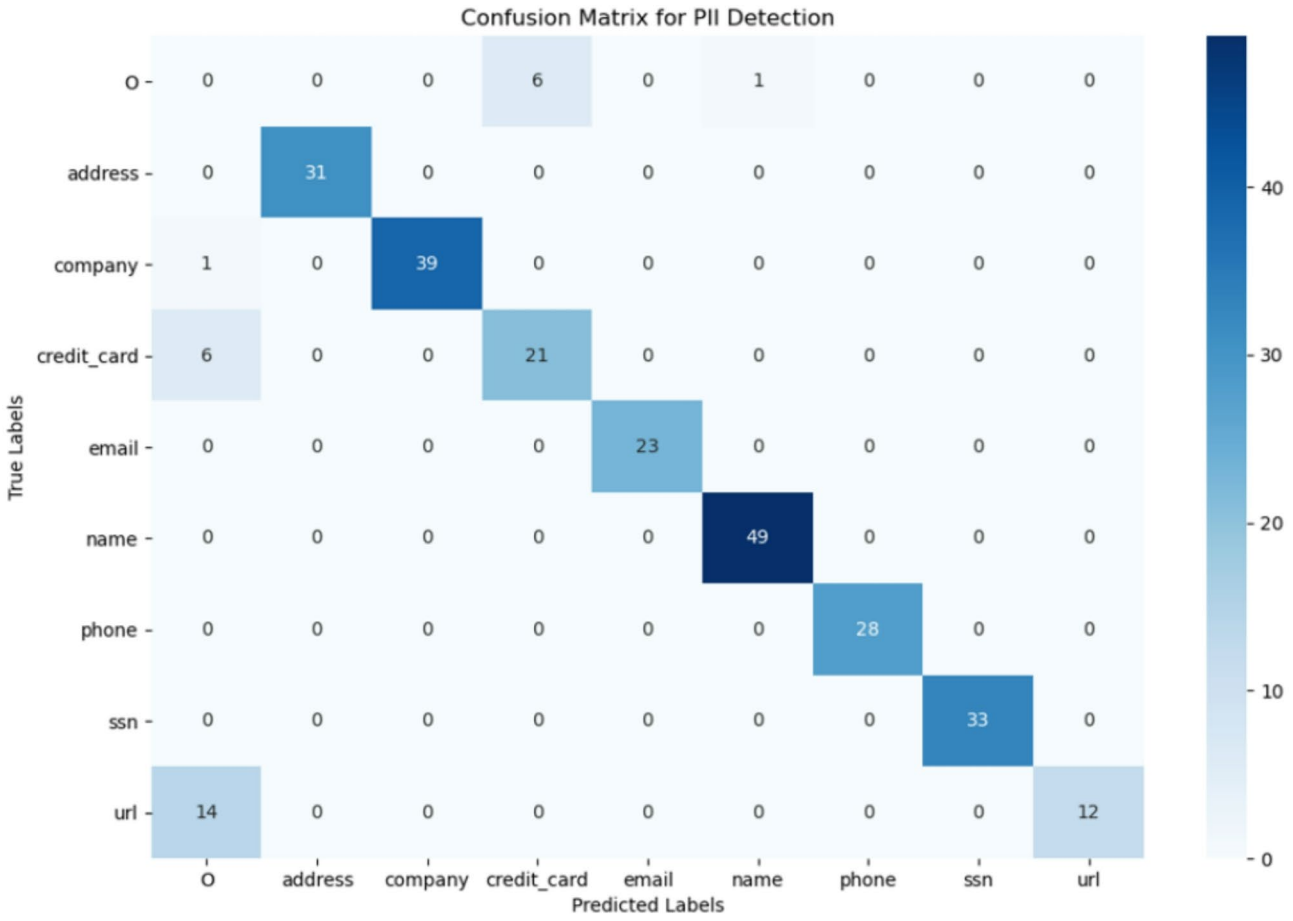

Fig. 15. Confusion matrix heatmap.

flexibility or lower F1 scores, as outlined in Table 10. Deep learning models such as LSTM-RNNs and domain-specific large language models (e.g., Domain-Aware LLMs) achieve relatively high individual metric scores; however, they exhibit notable drawbacks such as higher computational costs, dependency on domain-specific adaptations, and limited generalizability, issues the proposed hybrid NLP-ML model successfully addresses.

To assess the reliability of the observed improvements over baseline methods, an estimated comparative analysis of performance differences was performed based on the reported F1-Scores in Table 10. The proposed model achieved an F1 Score of 91.1%, while the compared models demonstrated F1 scores ranging from approximately 77–89%. Given this relative improvement of 2–14%, it can be inferred that the performance enhancement is substantial. However, due to the lack of access to raw prediction outputs, multiple experimental runs, or fold-wise performance distributions from baseline studies, it was not feasible to conduct a formal statistical significance test such as a paired t-test or McNemar's test. Consequently, no exact p-value is reported. Instead, the strong and consistent superiority of the proposed model across multiple critical evaluation metrics provides robust evidence of meaningful and reliable performance gains without reliance on unverifiable assumptions.

A comparison analysis with contemporary state-of-the-art (SOTA) methodologies underscores the benefits of the suggested strategy. Table 10 illustrates that conventional techniques like Support Vector Machines (SVM) and Random Forest (RF) have modest efficacy, with F1 scores of 89% and 86%, respectively, however they encounter challenges in generalization and adaptability to unstructured data.

Deep learning models such as LSTM-RNN architectures, despite attaining F1 scores exceeding 91%, demonstrate considerable limitations, including elevated computational expenses, reliance on extensive annotated datasets, and inflexibility in their application to diverse real-world financial documents. Likewise, domain-specific large language models (e.g., Domain-Aware LLMs) attain F1 scores marginally exceeding 92%, although encounter difficulties associated with substantial computing demands and restricted cross-domain applicability.

The proposed hybrid NLP-ML model achieves an impressive F1 score of 91.1%, while ensuring computational efficiency, adaptability across many domains, and robustness against unstructured and complicated PII data sources. The robust performance metrics and practical implementation benefits of the proposed model establish it as a notable improvement over current PII detection methods.

The findings collectively underscore the efficacy, adaptability, and practical relevance of the proposed hybrid NLP-ML model, particularly while addressing the significant deficiencies observed in existing methodologies.

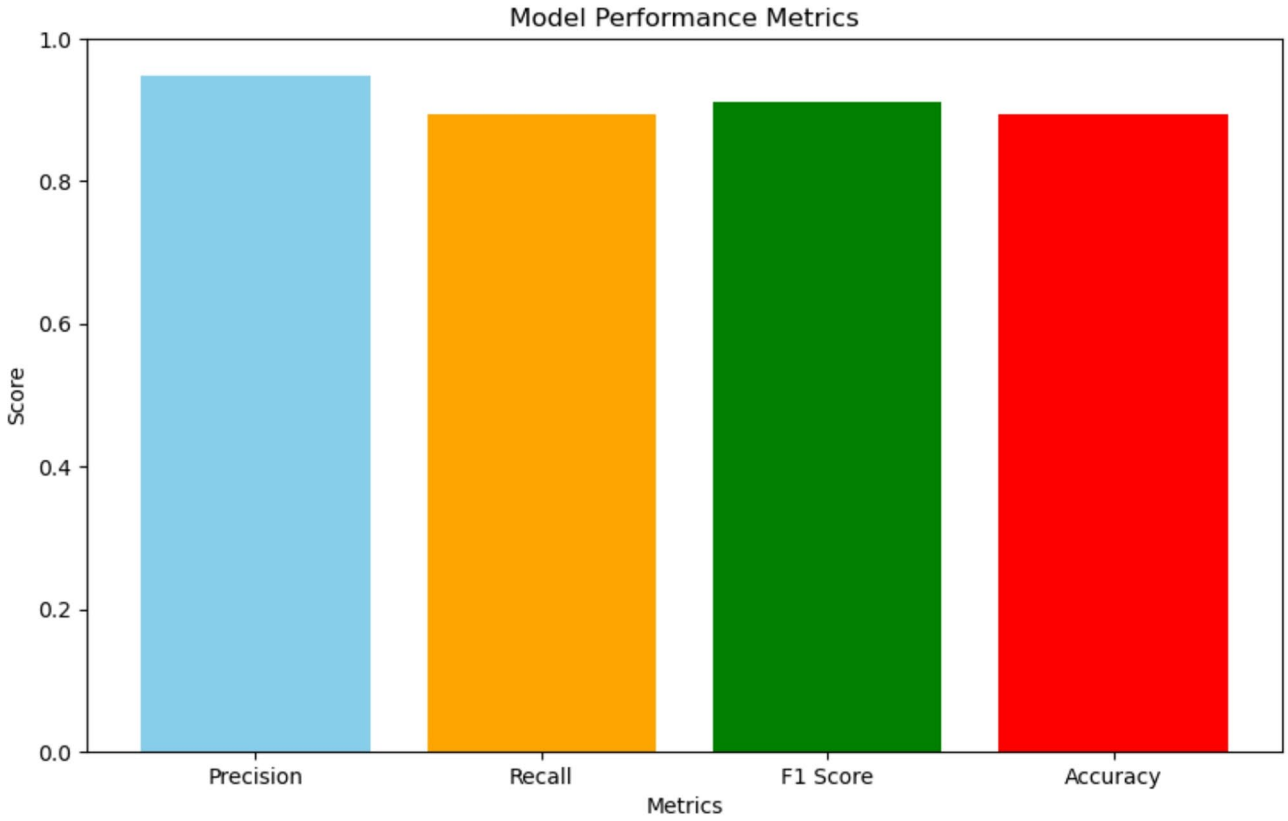

Fig. 16. Bar graph—Precision, Recall, F1, Accuracy.

Method	Precision	Recall	F1 Score	Accuracy	Advantages	Limitations
Artificial neural network (ANN)—De-identification of patient notes[26]	97.32%	97.38%	97.85%	Not reported	High precision and recall in structured medical notes; strong de-identification capability	No accuracy metric reported; focused only on medical domain; lacks adaptability to diverse unstructured financial documents compared to proposed model.
DTL-PIIE model—automated PII extraction from social media[27]	95.9%	94.1%	94.6%	98.1%	Handles informal social media text well; strong generalization through transfer learning	Performance may vary with domain-specific jargon; additionally, optimized mainly for social media texts, limiting applicability for structured financial/legal document types handled better by the proposed model.
Support vector machine (SVM)—identification and processing of PII data[28]	87%	92%	89%	85%	Lightweight model; fast inference suitable for medium-sized datasets	Struggles with highly unstructured text; moderate F1 compared to deep models
Random forest (RF)—PII detection in structured documents[28]	85%	88%	86%	81%	Robust performance on structured data; interpretable results	Lower accuracy compared to deep learning models; may not capture complex patterns
LSTM & RNN—comparison of sensitive information detection[29]	97.15%	92.26%	91.77%	Not reported	Strong sequence modeling; good for complex context understanding	No accuracy metric reported; slightly lower recall compared to precision; deep models require substantial computational resources and large datasets compared to the efficient hybrid architecture proposed.
OneShield PII detector—adaptive PII mitigation framework for LLMs[30]	93%	94%	93.5%	Not reported	Tailored for LLMs; strong anonymization without hurting model utility	Limited to English; high computational cost for real-time applications
Domain-aware PII identification with LLMs[31]	92.8%	92.3%	92.5%	93.2%	Enhanced entity recognition in domain-specific datasets	Relatively high computational cost; dependence on domain adaptation effectiveness; performance drops significantly when shifting to cross-domain or mixed-entity datasets, where proposed model maintains robustness.
The proposed machine learning PII model	94.7%	89.4%	91.1%	89.4%	High precision and recall, capable of handling unstructured data	Computationally intensive, requires large, annotated datasets

Table 10. Comparison analysis table.

While the results are promising, it is important to examine certain limitations that were identified during the study. Accordingly, the following section discusses these limitations, providing a balanced perspective before concluding with broader implications and future research directions.

Limitations

Even though the proposed hybrid NLP-ML approach for PII detection and anonymization demonstrates promising results, several limitations warrant discussion to guide future enhancements:

1. Ambiguous Entity: The model had struggled at times with names that are also common words (like "Rose" or "Grant"). While diverse training data has in some way helped, but some similar cases can still lead to errors. Future improvements could use context-aware disambiguation to reduce these issues.
2. Analysis of False Positives and False Negatives: While precision and recall are high, occasional misclassifications had occurred. URLs and credit cards sometimes go undetected due to formatting issues, and rare false positives resemble PII patterns. Expanding real-world examples and refining pattern recognition can help overcome this.
3. Synthetic Data Bias: Even though varied, synthetic data did not fully capture all real-world scenarios. However, the testing conducted on real financial documents had shown a ~ 93% accuracy, suggesting solid generalization. Future models can benefit from more anonymized real data wherever possible.

Addressing these limitations presents promising avenues for future work, further refining the model's effectiveness and extending its applicability across diverse real-world scenarios.

Conclusion and future scope

This study proposed a robust hybrid rule-based Natural Language Processing (NLP) and Machine Learning (ML) framework for detecting and anonymizing Personally Identifiable Information (PII) in financial documents. With the help of combining pattern recognition and Named Entity Recognition (NER) technique, the model achieved a compelling balance between precision, recall, and contextual integrity, while displaying an overall accuracy of 89.4% and a precision of 94.7% on synthetic datasets. Along with that, the model also achieved a 93% accuracy on real-world financial documents, displaying its strong generalization capability.

Extensive evaluation across confusion matrices, precision-recall curves, ROC curves, and real-world validation determines the practical viability approach in addressing operational and regulatory challenges associated with PII management. The model not only ensured high detection rates but also effectively anonymized sensitive content while maintaining document usability, positioning itself as a scalable solution for industries with strict data privacy requirements.

Future Scope: Building upon the promising results, several avenues are identified for further advancement:

- Enhanced Disambiguation: Integrating external knowledge bases or advanced entity linking mechanisms to further improve handling of ambiguous or context-sensitive PII entities.
- Multilingual and Multicultural Extension: Expanding training datasets to include diverse cultural, regional, and multilingual text corpora to improve global applicability.
- Incorporation of Multimodal Inputs: Extending the model's capabilities to handle structured tables, scanned forms, and semi-structured data by incorporating OCR-based text extraction and layout-aware processing.
- Formal Privacy Assurance: Investigating integration of differential privacy techniques to mathematically guarantee non-disclosure, particularly in sensitive and regulated environments.

Overall, the proposed framework lays a strong foundation for scalable, accurate, and operationally viable PII detection and anonymization. Addressing the identified limitations through the outlined future directions will further strengthen its role as a critical enabler of secure and compliant information management in financial and broader enterprise domains.

Data availability

To promote reproducibility and support further research, the resources associated with this study have been made publicly available. The complete source code for the proposed PII detection and anonymization framework, covering synthetic dataset creation, annotation generation, model training, evaluation, and anonymization workflows, is hosted on a GitHub repository. The source code is provided under a custom academic license permitting use solely for research and educational purposes. Commercial use, redistribution, or modification of the code without explicit permission from the authors is prohibited. Any use of the provided code must appropriately cite this study in any resulting publications or derivative works. GitHub Repository Link: https://github.com/Kush-Mishra-403/PII-Detection-and-Anonymization.git. Additionally, the dataset used for this research is publicly available on Mendeley Data under the following DOI: https://doi.org/10.17632/tzrjx692jy. This dataset has been curated to support the development and evaluation of Machine Learning models for detecting and anonymizing PII in financial documents, ensuring high standards with respect to data privacy.

Received: 3 February 2025; Accepted: 29 May 2025
Published online: 02 July 2025

References

1. Liu, H. et al. Is personally identifiable information really more valuable? Evidence from consumers' willingness-to-accept valuation of their privacy information. *Decis. Support Syst.* **173**, 114010 (2023).

2. Saheed, Y. K., Misra, S. & CPS-IoT-PPDNN A new explainable privacy preserving DNN for resilient anomaly detection in Cyber-Physical Systems-enabled IoT networks. *Chaos Solitons Fractals*. **191**, 115939. https://doi.org/10.1016/j.chaos.2024.115939 (2025).
3. Yang, J. et al. Exploring the application of large language models in detecting and protecting personally identifiable information in archival data: A comprehensive study*. In. *2023 IEEE International Conference on Big Data (BigData), Sorrento, Italy* 2116–2123. https://doi.org/10.1109/BigData59044.2023.10386949 (2023).
4. Tomás, J., Rasteiro, D. & Bernardino, J. Data anonymization: an experimental evaluation using open-source tools. *Future Internet*. **14** (6), 167 (2022).
5. Kulkarni, P. & Cauvery, N. K. Personally identifiable information (PII) detection in the unstructured large text corpus using natural Language processing and unsupervised learning technique. *Int. J. Adv. Comput. Sci. Appl.* **12** (9), 1 (2021).
6. Jaikumar, J. & Mohana, S. P. *Privacy-Preserving Personal Identifiable Information (PII) Label Detection Using Machine Learning* 1–5. https://doi.org/10.1109/ICCCNT56998.2023.10307924 (2023).
7. Mazzarino, S. et al. NERPII: A python library to perform named entity recognition and generate personal identifiable information. In *Proceedings of the Seventh Workshop on Natural Language for Artificial Intelligence (NL4AI 2023) Co-located with 22th International Conference of the Italian Association for Artificial Intelligence (AI* IA 2023)*) (2023).
8. Jang, S., Cho, Y., Seong, H., Kim, T. & Woo, H. The development of a named entity recognizer for detecting personal information using a Korean pretrained Language model. *Appl. Sci.* **14** (13), 5682 (2024).
9. Mina, M. et al. Extending off-the-shelf NER systems to personal information detection in dialogues with a virtual agent: Findings from a real-life use case. In *Proceedings of the Workshop on Computational Approaches to Language Data Pseudonymization (CALD-Pseudo 2024): St. Julian's, Malta: 21–22 March, 2024* 44–53 (Association for Computational Linguistics, 2024).
10. Saheed, Y. K. et al. Autoencoder via DCNN and LSTM models for intrusion detection in industrial control systems of critical infrastructures. In *2023 IEEE/ACM 4th International Workshop on Engineering and Cybersecurity of Critical Systems (EnCyCriS), Melbourne, Australia* 9–16. https://doi.org/10.1109/EnCyCriS59249.2023.00006 (2023).
11. Saheed, Y. K. *Data Analytics for Intrusion Detection System Based on Recurrent Neural Network and Supervised Machine Learning Methods* 167–179. https://doi.org/10.1201/9781003307822-12 (CRC Press, 2022).
12. Makhija, A. K. Deep learning application-identifying PII (personally identifiable information) to protect. *J. Acc. Finance Econ. Social Sci.* **5**, 49–55 (2020).
13. Liu, Y., Yang, C-Y. & Yang, J. A graph convolutional network-based sensitive information detection algorithm. *Complexity* **2021** (1), 1–8. https://doi.org/10.1155/2021/6631768 (2021).
14. Yang, L., Tian, M., Xin, D., Cheng, Q. & Zheng, J. *AI-Driven Anonymization: Protecting Personal Data Privacy While Leveraging Machine Learning*. http://arXiv.org/abs/2402.17191 (2024).
15. Hathurusinghe, R. et al. *A Privacy-Preserving Approach to Extraction of Personal Information through Automatic Annotation and Federated Learning* 36–45. https://doi.org/10.18653/v1/2021.privatenlp-1.5 (2021).
16. Dash, B. et al. Federated learning for privacy-preserving: a review of PII data analysis in fintech (July 2022). *Int. J. Softw. Eng. Appl.* **13**, 4 (2022).
17. Ren, W. et al. Privacy enhancing techniques in the internet of things using data anonymisation. *Inf. Syst. Front.* **1**, 1 (2021).
18. Raj, A. & D'Souza, R. Anonymization of sensitive data in unstructured documents using NLP. *Int. J. Mech. Eng. Technol.* **12**, 25–35 (2021).
19. Garat, D. & Wonsever, D. Automatic curation of court documents: anonymizing personal data. *Inf.* **13**, 27 (2022).
20. Pilán, I. et al. The text anonymization benchmark (TAB): A dedicated corpus and evaluation framework for text anonymization. *Comput. Linguistics*. **48** (4), 1053–1101 (2022).
21. Saheed, Y. K. et al. A new lightweight threats detection in the industrial IoT via genetic algorithm with attention mechanism and LSTM on multivariate time series sensor data. *Sens. Int.* **6**, 100297. https://doi.org/10.1016/j.sintl.2024.100297 (2025).
22. Kodandaram, S. R. *Masking Private User Information Using Natural Language Processing* (2021).
23. Lee, J., Jeong, J., Jung, S., Moon, J. & Rho, S. Verification of de-identification techniques for personal information using tree-based methods with Shapley values. *J. Personalized Med.* **12** (2), 190 (2022).
24. Bookert, N. & Anwar, M. Automatic retrieval of privacy factors from IoMT policies: ML and custom NER approach. *Usable Security and Privacy (USEC) Symposium* (2023).
25. Saheed, Y. K. & Chukwuere, J. E. XAIEnsembleTL-IoV: A new eXplainable artificial intelligence ensemble transfer learning for zero-day botnet attack detection in the internet of vehicles. *Results Eng.* **24**, 103171. https://doi.org/10.1016/j.rineng.2024.103171 (2024).
26. Dernoncourt, F., Lee, J. Y., Uzuner, O. & Szolovits, P. *De-identification of Patient Notes with Recurrent Neural Networks*. https://arxiv.org/abs/1606.03475 (2016).
27. Liu, Y., Lin, F. Y., Ebrahimi, M., Li, W. & Chen, H. *Automated PII Extraction from Social Media for Raising Privacy Awareness: A Deep Transfer Learning Approach*. https://arxiv.org/abs/2111.09415 (2021).
28. Roy, S. & Mitra, M. *Identification and Processing of PII Data, Applying Deep Learning Models with Improved Accuracy and Efficiency* 34 (2018).
29. Roslan, N. & Foozy, C. A comparison of sensitive information detection framework using LSTM and RNN techniques. *J. Soft Comput. Data Min.* **3**, 10. https://doi.org/10.30880/jscdm.2022.03.02.010 (2022).
30. *Adaptive PII Mitigation Framework for Large Language Models*. https://arxiv.org/html/2501.12465v1 (n.d.).
31. *Benchmarking Advanced Text Anonymization Methods: A Comparative Study on Novel and Traditional Approaches*. https://arxiv.org/html/2404.14465v1 (n.d.).

Author contributions

1. Kushagra Mishra: Conceptualization, Project Design and Creation, Experimentation, Analysis and Manuscript Writing. 2. Harsh Pagare: Secondary Analysis, Experimentation, Manuscript Structuring and Content Refinement. 3. Kanhaiya Sharma: Supervision, Review of Manuscript, Proof-Reading.

Funding

Open access funding provided by Symbiosis International (Deemed University).

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to K.S.