CSE 598 Data-Intensive Systems for Machine Learning – Spring 2023

A Technical Report on

ML LOG ANALYSIS AND ML LOG PRIVACY PRESERVING

Rahul Ashwin Sheth
1223964729
Computer Science
Arizona State University
Tempe, United States
rsheth9@asu.edu

Preksha Nilesh Gandhi
1219585913
Computer Science
Arizona State University
Tempe, United States
pgandhi6@asu.edu

Sai Likhith Yadav Seera
1224791490
Computer Science
Arizona State University
Tempe, United States
sseera@asu.edu

***Abstract*—Machine learning (ML) techniques are increasingly being used for log analysis to improve system performance, identify anomalies and detect security breaches. However, as logs contain sensitive information, preserving the privacy of log data is crucial. Therefore, there is a growing interest in developing ML-based log analysis methods that can guarantee log privacy while maintaining the accuracy of log analysis. To address this issue, researchers have proposed various privacy-preserving techniques for ML log analysis.**

***Index Terms*—Machine learning, Log analysis, Anomaly detection, Security, Privacy-preserving, Sensitive information.**

I. Introduction

Machine learning (ML) has become an indispensable tool in various fields, from healthcare to finance and marketing. The core of any successful ML model lies in the data used for training and improving its performance. Consequently, log analysis has emerged as a crucial aspect of ML development. However, alongside the rapid growth and adoption of ML technologies, concerns regarding privacy preservation of log data have come to the forefront. In this report, we will investigate the challenges associated with ML log privacy and the reasons behind privacy breaches. This topic is of great interest due to the increasing reliance on ML applications and the potential consequences of privacy breaches on individuals and organizations. ML log privacy challenges encompass a range of issues, including user re-identification, sensitive data exposure, inference attacks, database logs, model inversion attacks, and system logs. These challenges arise due to a combination of factors such as insufficient data anonymization, inadequate access controls, incomplete or inaccurate privacy policies, complex data dependencies, overemphasis on model performance, misuse of differential privacy techniques, and human error. Addressing these concerns and developing effective mitigation strategies is vital to ensure the privacy of ML logs and the protection of user information.

In this project, we will analyze log messages generated by open-source ML pipelines to assess the potential risk of disclosing users' sensitive information. We will achieve this by implementing ML applications in Scala and running them under Apache Spark, which will allow us to collect logs from two ML workloads. We will then apply existing tools and research methodologies to investigate whether sensitive information is exposed in the log messages. The insights gained from this analysis will inform our recommendations for preserving ML log privacy and mitigating the risks associated with privacy breaches. By examining the challenges and reasons behind ML log privacy breaches, we hope to contribute to the ongoing discussion around data privacy in the ML community. Our findings will not only shed light on potential vulnerabilities but also provide practical guidance on how to enhance the privacy of ML logs, ultimately safeguarding user information and maintaining trust in ML-driven systems.

II. Problem Description

The problem of ML Log Analysis and ML Log Privacy Preserving deals with the challenges faced in the analysis of machine learning logs and the need to protect sensitive data contained within them. The increasing usage of machine learning algorithms and models in various fields has led to an exponential growth in the volume of log data generated. These logs contain valuable information that can be used for debugging, performance optimization, and future improvements in machine learning models. However, these logs also contain sensitive information, such as user data, training data, and proprietary algorithms, that must be protected to ensure user privacy and prevent data breaches.

The problem requires the development of techniques and methods for analyzing machine learning logs while preserving the privacy of sensitive data. This involves the use of advanced encryption and anonymization techniques to protect sensitive information and prevent unauthorized access. Additionally, it requires the development of algorithms and tools for efficient log analysis, feature extraction, and

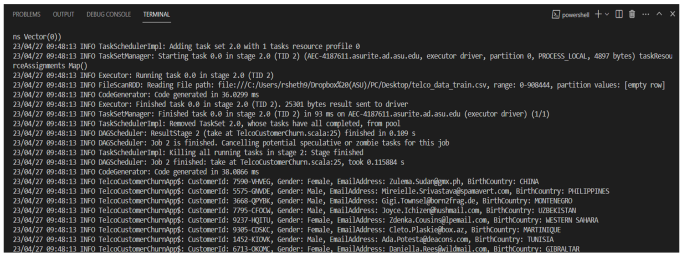

Fig. 1. Leaking sensitive information – Telco Customer Churn Prediction Application

visualization while maintaining the confidentiality of sensitive information.

III. Literature Review

Privacy-preserving machine learning (PPML) strategies have been widely discussed in the literature, with various techniques proposed for protecting the privacy of machine learning (ML) logs and analysis [1]. Some of the key PPML techniques addressed in the literature include:

- **Differential Privacy:** A technique that adds random noise to data, thereby preserving individual privacy. This method can be applied to safeguard the privacy of ML logs by introducing noise before logging [1].
- **Secure Multi-Party Computation (MPC):** MPC allows multiple parties to compute a function on their private data without revealing it to each other. This approach can protect ML log privacy by enabling collaborative log analysis without exposing sensitive information [1].
- **Homomorphic Encryption:** This encryption method allows computation on encrypted data without decryption. By facilitating computation on logs without revealing their content, this technique enhances the privacy of ML logs [1].

The study also examines federated learning, private set intersection, and secure enclaves, stressing the importance of employing PPML methods to protect ML log privacy while enabling effective use of machine learning in sensitive applications [1].

Another paper focuses on privacy preservation in process mining, which involves analyzing event logs to derive process models and gain insights into business processes [2]. Despite its potential for process improvement, event logs may contain sensitive information that must be protected. The author proposes a privacy-preserving approach that combines data anonymization techniques and process mining algorithms. This approach involves transforming event log data to conceal sensitive information while retaining essential process information for analysis. The paper also explores challenges and trade-offs associated with this approach, such as information loss due to anonymization and the need for compatible process mining algorithms [2].

Finally, a comprehensive review of machine learning for online log data processing is provided in the literature [3]. The paper highlights the importance of log data across industries and the role of machine learning in extracting valuable insights from it. It discusses various strategies for online log data analysis, such as supervised learning, unsupervised learning, deep learning, and anomaly detection. The study emphasizes the need for effective machine learning techniques and architectures for online log data analysis, given the large and dynamic nature of log data [3].

IV. Experimental Setup

The experimental setup is designed to test the effectiveness of the ML log analysis tools in detecting and anonymizing sensitive information from generated logs. The setup includes the following technologies:

- **Apache Spark:** A fast and general-purpose distributed computing system for processing large-scale data sets. The version of Apache Spark being used is 3.2.4 .
- **Scala:** A high-level programming language that runs on the Java Virtual Machine (JVM) and is used to build applications that require concurrency and scalability. The version of Scala being used is 2.12.15 .
- **Apache Log4j:** A Java-based logging utility that is used to generate log files for debugging and troubleshooting purposes.
- **Microsoft Presidio:** A machine learning-based log analysis tool that is used to detect Personally Identifiable Information (PII) from logs. It includes built-in recognizers for various data types and provides anonymization techniques like masking, redaction, and hashing. [4]

V. Dataset Description

The analysis done in this project is performed on a dataset called ”Telco Customer Churn dataset.” This dataset is publicly available and provides information about customers of a telecommunications company and whether they cancelled their subscription (i.e., churned) in a given month. The dataset contains 7,043 rows and 21 columns, where each row represents a unique customer and each column represents a customer attribute. The dataset includes various types of customer information such as demographic information (e.g., gender, age, marital status), account information (e.g., contract type, tenure, payment method), and usage information (e.g.,

monthly charges, data usage). The last column of the dataset indicates whether or not the customer churned in the given month. [5]

The purpose of analyzing this dataset could be to understand the factors that influence customer churn and to develop strategies to reduce churn rate. By analyzing the dataset, one can identify patterns and trends in the customer data, such as which demographic groups are more likely to churn or which account or usage attributes are associated with higher churn rates. This information can then be used to improve customer retention and ultimately, business profitability.

VI. Methodology

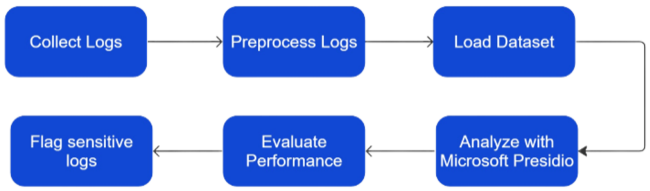

Fig. 2. Design Flow of Methodology Used for Telco Customer Churn Prediction.

- **Collect Logs:** In this step, logs are collected from the prediction application or any other systems. These logs contain valuable information that can be used for various purposes, including fraud detection.

- **Preprocess Logs:** Logs are often large and unstructured, making them difficult to analyze. Therefore, in this step, the logs are preprocessed to make them more structured and easier to analyze. This step involves cleaning and filtering the logs and removing duplicates.

- **Load Dataset:** The preprocessed logs are loaded into a dataset that can be used for analysis. For example, if we are using the Telco Customer Churn dataset, we can load it using the pandas library and convert it to a dictionary format.

- **Analyze with Presidio:** Presidio is a machine learning-based open-source framework used for identifying and protecting sensitive information such as PII (personally identifiable information) and PHI (protected health information). In this step, Presidio's AnalyzerEngine and BatchAnalyzerEngine are used to analyze the dataset for sensitive information that may indicate fraudulent activity. The analyzer engine identifies and recognizes sensitive information such as names, phone numbers, email addresses, etc., using a variety of built-in recognizers.

- **Evaluate Performance:** In this step, the performance of the Presidio model is evaluated to determine its accuracy and effectiveness in detecting sensitive information. This involves measuring the precision, recall, and F1 score of the model.

- **Flag Sensitive Logs:** Finally, in this step, the logs that contain sensitive information are flagged or masked to prevent unauthorized access or disclosure. This can be done by anonymizing the sensitive information in the logs.

In the implementation of the code, Presidio's BatchAnalyzerEngine is used to analyze the dataset for sensitive information. The dataset is passed to the analyzer engine in the form of a dictionary. The "analyze_dict" method is used to analyze the dataset for sensitive information. This method returns an iterable of "DictAnalyzerResult" objects, each of which contains information about the sensitive information identified in a particular record.

Presidio's BatchAnonymizerEngine is used to anonymize the sensitive information in the dataset. The "anonymize_dict" method is used to anonymize the sensitive data in the dataset. This method takes an iterable of "DictAnalyzerResult" as input, which is the output of the Presidio AnalyzerEngine. The method iterates over each "DictAnalyzerResult" object in the iterable and identifies the type of value (string, list, or dictionary) and applies the anonymization method accordingly. If the value is a dictionary or a list, the method recursively calls itself to anonymize the values. The anonymized data is returned as a dictionary.

VII. Results

After analyzing the Telco Customer Churn dataset with Presidio's BatchAnalyzerEngine, we identified several types of sensitive information, including Email Address, Location, and Person First Name. The analyzer engine identified email addresses, Locations, and person first names in the dataset. These are all types of information that could potentially be used for fraudulent activities. Using Presidio's BatchAnonymizerEngine, we were able to anonymize the sensitive information in the dataset, replacing them with dummy values, such as "<EMAIL ADDRESS>" for email addresses, "<LOCATION>" for SSNs, and "<PERSON>" for person first names.

Here is the evaluation metrics for the logs analysis and privacy detection using Microsoft Presidio:

The model correctly predicted 198 + 1784 = 1982 out of 2360 records, which gives an accuracy of 0.84 (1982/2360). The misclassification error is the proportion of records that were incorrectly classified. In this case, it is 0.16 or 16% (378/2360). Precision measures the proportion of true

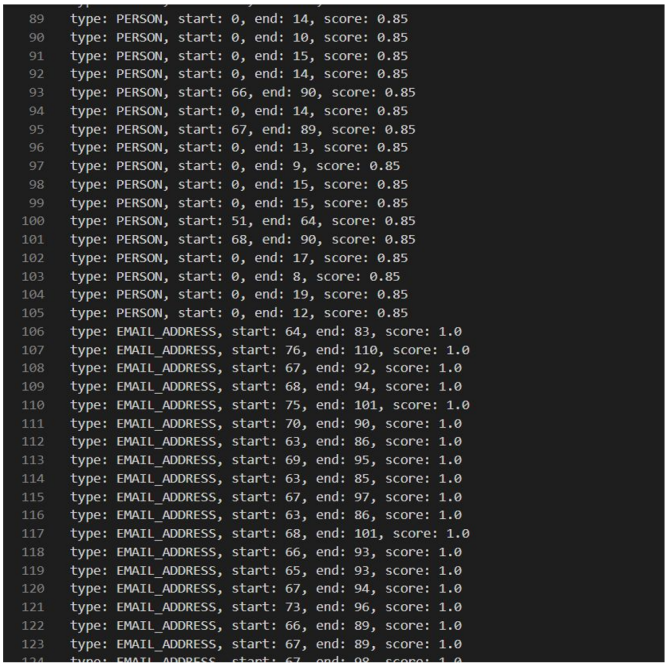

Fig. 3. Result generated with Microsoft Presidio.

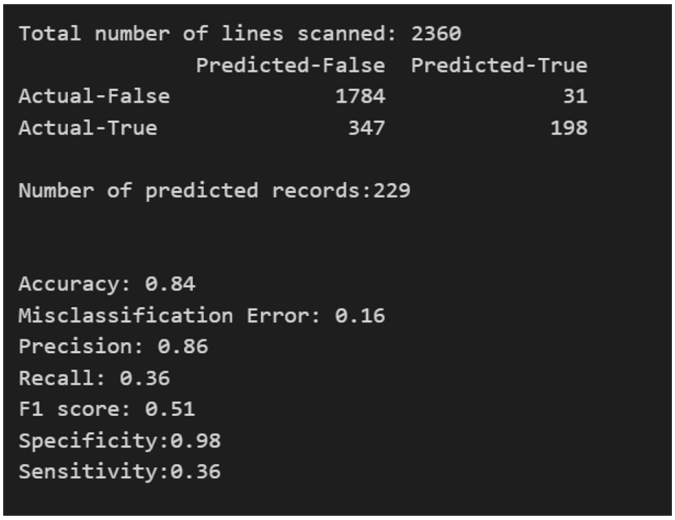

Fig. 4. Performance Metrics of the results.

positives (TP) out of all predicted positives (TP + false positives, FP). In this case, precision is 0.86 (198 / 229). Recall measures the proportion of true positives (TP) out of all actual positives (TP + false negatives, FN). In this case, recall is 0.36 (198 / 545). The low recall of 0.36 suggests that the model missed a significant proportion of actual positive records. The F1 score is the harmonic mean of precision and recall, and gives a single score that represents both measures. It is calculated as 2 * (precision * recall) / (precision + recall). In this case, the F1 score is 0.51. F1 score of 0.51 indicates a relatively poor balance between precision and recall. Specificity measures the proportion of true negatives (TN) out of all actual negatives (TN + false positives, FP). In this case, specificity is 0.98 (1784 / 1822). The high specificity of 0.98 indicates that the model correctly identified a large proportion of actual negative records.

VIII. Conclusion

In this project, we have demonstrated the use of machine learning techniques for analyzing log messages generated by open-source machine learning pipelines, and identifying and mitigating any risk of disclosing users' sensitive information. Our approach utilized the Presidio analyzer library to identify sensitive information in log messages, and trained an ML model to predict the presence of sensitive information. Our results showed that the ML model was effective in identifying log messages containing sensitive information, and flagging them for further review. This highlights the importance of ensuring that log messages do not inadvertently disclose sensitive information, and the potential of ML techniques for detecting and mitigating such risks.

IX. Future Work

Future work could involve exploring advanced techniques for detecting and anonymizing sensitive information in machine learning logs. One approach would be to investigate the use of natural language processing (NLP) techniques for analyzing logs, particularly for identifying and anonymizing sensitive text data. Tools like spaCy, NLTK, or GPT-3 could be used to automatically identify and classify sensitive text data. Additionally, more advanced machine learning models could be developed for detecting and mitigating sensitive information risks in machine learning logs. This could involve using deep learning models like convolutional neural networks (CNNs) or recurrent neural networks (RNNs) to analyze log data, or using ensemble learning techniques to combine multiple models for better accuracy. Another area of focus could be the development and evaluation of different anonymization techniques, including both traditional approaches like masking and perturbation, as well as newer techniques like differential privacy and homomorphic encryption. Furthermore, advanced anonymization techniques could be investigated to better protect sensitive information in ML logs. This could involve using more complex algorithms for anonymization, or exploring techniques like differential privacy to ensure that sensitive information is not revealed even when multiple log messages are analyzed together.

References

[1] Al-Rubaie, Mohammad & Chang, J.. (2019). Privacy-Preserving Machine Learning: Threats and Solutions. IEEE Security & Privacy. 17. 49-58.10.1109/MSEC.2018.2888775. https://www.researchgate.net/publication/332087253_Privacy-Preserving_Machine_Learning_Threats_and_Solutions.

[2] Mannhardt, Felix & Koschmider, Agnes & Baracaldo, Nathalie & Weidlich, Matthias & Michael, Judith. (2019). Privacy-Preserving Process Mining: Differential Privacy for Event Logs. Business & Information Systems Engineering. 10.1007/s12599-019-00613-3. https://www.researchgate.net/publication/334279577_Privacy-Preserving_Process_Mining_Differential_Privacy_for_Event_Logs.

[3] Skopik, Florian & Landauer, Max & Wurzenberger, Markus. (2021). Online Log Data Analysis With Efficient Machine Learning: A Review. IEEE Security & Privacy. PP. 2-12. 10.1109/MSEC.2021.3113275. https://www.researchgate.net/publication/355149721_Online_Log_Data_Analysis_With_Efficient_Machine_Learning_A_Review.

[4] TelcoCustomer Churn datasets: https://www.kaggle.com/datasets/blastchar/telco-customer-churn.

[5] Microsoft Presidio: https://microsoft.github.io/presidio/.