Library Progress International
Vol.44 No.3, Jul-Dec 2024: P.19826-19837

Print version ISSN 0970 1052
Online version ISSN 2320 317X

Original Article

Available online at www.bpasjournals.com

PPRF: Privacy Preserving Random Forest for Data Stream Mining

Aniket Patel[1], Dr. Kiran Amin[2]

[1]Department of Computer Science and Engineering, Ganpat University, Gujarat, India
[2]Department of Computer Engineering, Ganpat University, Gujarat, India
[1]arp02@ganpatuniversity.ac.in
[2]kiran.amin@ganpatuniversity.ac.in

How to cite this article: Aniket Patel,Kiran Amin (2024) PPRF: Privacy Preserving Random Forest for Data Stream Mining. *Library Progress International,* 44(3), 19826-19837

ABSTRACT

In today's technology-driven landscape, the perva- sive use of online services across diverse domains has to led the generation of vast datasets, necessitating advanced data mining techniques for meaningful insights. The advent of data streams, characterized by continuous and dynamic data flows, presents a significant challenge, prompting the evolution of data stream mining. This field addresses issues such as rapid changes in streaming data and the need for quick algorithms. To tackle these challenges, an innovative approach named (Effective Privacy Preserving eXtreme Gradient Boosting) PPRF is proposed, combining Adaptive XGBOOST for continuous learning from evolving data streams with PPXGBOOST for privacy preservation..

1. INTRODUCTION

Machine learning (ML) applications like remote healthcare and activity identification have gotten much press as a huge step forward in the field. Data streams describe these ML applications: data is generated by numerous devices and typically arrives on time. Because data may come quickly, the data stream should be handled effectively on-the-fly for either training the ML model or inferring (i.e., predicting or assessing) an unlabeled instance by the model. The Hoeffding Tree (H.T.) model [1], a variant of the decision tree model, [2]has become the industry standard for data stream processing. Recently, the adaptive XGBOOST method is also proposed for evolving data streams [3].

Outsourcing H.T. training and inference to cloud platforms [4] [2] is a potential solution for effectively processing data streams. However, data privacy and model secrecy are seriously endangered due to this. Therefore, all data samples, the model, the inference output, and any intermediate data created during model training and inference should be shielded from the Cloud Service Provider for privacy-sensitive applications, such as in the healthcare domain (CSP). When training an H.T., the key task is to categorize each new data sample according to the current tree and count the frequency of distinct feature values.

Privacy-preserving data mining (PPDM) uses cryptographic primitives like Secure Multi-Party Computation (SMC) [5] [6] [7] [8] [9] [10] or Homomorphic Encryption (HE) [11] [12] [13] [14] to safeguard the privacy of outsourced ML activities. However, most existing PPDM approaches cannot be used to process data streams because they: cannot efficiently process complex functions such as logarithm and exponential operations, which are essential for H.T. model training; impose too high computation and communication overheads on clients, and leak statistical information and tree structures.

2. LITERATURE REVIEW

This section discusses mainly two aspects. One is data stream mining, and another is homomorphic encryption. There are several traditional techniques for data stream mining. This paper focuses on data stream mining using a machine learning case scenario. The main agenda for privacy preservation in data stream mining, particularly during machine learning, is a demanding area in today's scenario. Most companies are shifting their data to the public clouds. They use ML algorithms available with their cloud. It uses companies' data, train the model, and

provides you with the output. The main drawback is data privacy used for training on the public cloud. This paper provides the solution to that problem.

A. Data Stream mining

Huge amounts of transactional, digital, and sensory data are created daily and must be examined live as they come in. One of the primary sources of what is known as big data may be thought of as streaming data. The last 10 years have seen much interest in predictive modeling for data streams and big data; however, many research methodologies are frequently created for well-behaved controlled issue settings, ignoring significant constraints presented by real-world applications [15]. The following table summarizes various methods used for privacy-preserving data stream mining in machine learning.

Table 1Various privacy-preserving ML algorithms

Algorithm	Broad Category	Task	Does it work for Data Stream?	Does it deal with privacy?
Adaptive XGBoost [3]	**Ensemble**	**Classification**	**Yes**	**No**
Logistic Regression with Secret Sharing [5]	**Regression**	**Classification**	**No**	**Yes**
Gradient Boosting Decision Trees [11]	**Ensemble**	**Classification**	**NO**	**Yes**
Decision Tree [8]	**Decision Tree**	**Classification**	**NO**	**No**

B. Homomorphic Encryption

Homomorphic encryption is a kind of encryption that permits ciphertext computation. It produces a result that is encrypted. When decrypted, the outcome is identical to the original actions as though they were carried out on plaintext [16]. The aim of homomorphic encryption is to make it possible to compute encrypted data [2]. As a result, homomorphic encryption enables complicated mathematical computations on encrypted data without jeopardizing privacy.

3. PROPOSED SYSTEM

Since data comes from complex environment, may be noisy, redundant, contain outliers, and lack values, data preparation is a crucial step in all real-world data analysis applications. There are several well-known standard methods for preparing offline data; for an example, see [17]. However, the data stream situation poses fresh problems that should be assressed to implement the research in real world scenario.

This section provides proposed system structure. Section A explains the mathematical model for data stream handling. Section B provides proposed architecture. Section C provides proposed EEPXGBoost Algorithm while section D provides modified homomorphic encryption algorithm.

A. The mathematical model for Datastream handling

For example, a continuous data stream can be defined as D={($\bar{x}$t,yt)|t=1,..., T} where T→∞, where $\bar{x}$t is a feature vector, and yt would be the corresponding target. Instead of using the same data to select each base function fb, it uses subsamples of data obtained from non-overlapping windows. New data samples are stored in the buffer

B = ($\bar{x}$i,yi) : i ∈ {1, 2,3, ..., W} with size |w|=W samples

Once the buffer is full, AXGB will proceed to train a single function fs. Y^s=∑_(s=1)^s⁣ ⟦fs (ws)=Y⟧ ^((s-1)) +fs (ws)

The index s of the new base function are obtained. If it is the first ensemble member, f, then the data in the buffer is directly used. If s>1, the data is passed through the ensemble, and residuals from the first s-1 models in the ensemble are used to obtain the new base function fs.

To update the Ensemble (E), two strategies are i) PUSH Strategy and ii) REPLACEMENT Strategy. In the PUSH strategy, it works like Queue. When new models are created, they are appended to the Ensemble (E), as shown in the figure. On the other hand, if E is full, older modules are removed before appending a new model.

Figure 1. PUSH Strategy

While in the REPLACEMENT Strategy, older numbers are replaced with a new one. It replaces older data, as shown in the figure.

Figure 2. REPLACEMENT Strategy

The main requirement of stream learning is that models be ready to provide predictions.

Given the incremental nature of AXGB, if the window size is fixed, the ensemble will require S.W samples to create a full ensemble. Full ensemble=S×W samples. This technique has a problem; performance can be suboptimal at the beginning of the stream. So, to solve such a problem, the dynamic window size can be the best solution. Window (W) size doubles in each iteration from a given minimum size W_{min} until it is reached to $W_{max.}$ The window size (W_i) for i^{th} iteration can be defined as:-

$$W_{(i)}=\min(W_{min}.2^{i},W_{max})$$

So, several iterations I to reach the max window size are:

$$i=\left\lceil \log_2\left(\frac{W_{max}}{W_{min}}\right)\right\rceil$$

B. Proposed Architecture

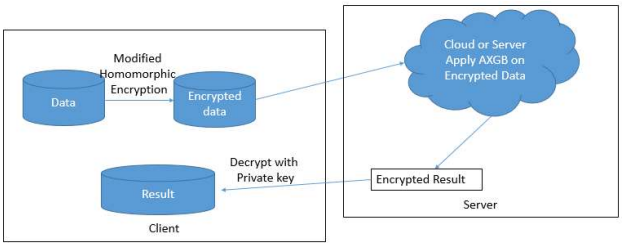

Figure 3. Proposed architecture

- Use either S.I. (MKS) or CGS as primary units. (S.I. units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be using English units as identifiers in trade, such as "3.5-inch disk drive".
- Avoid combining S.I. and CGS units, such as current in amperes and magnetic field in oersteds. It often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity in an equation.

- Do not mix complete spellings and abbreviations of units: "Wb/m2" or "webers per square meter," not "webers/m2". Instead, spell out units when they appear in text: ". . . a few henries", not ". . . a few H".
- Use a zero before decimal points: "0.25", not ".25". Use "cm3", not "cc." (*bullet list*)

C. Proposed PPRF Algorithm

PPRF Algorithm encrypts the data before applying the ML model. Then ML model is applied to encrypted data, and results are generated in the encrypted form; this result is sent to the client. The client can decrypt the result; thus, it provides very good security. So, PPRF = ADaptiveXGBOOST + Modified Homomorphic Encryption + OPE (Order Preserving Encryption)

- Step1: Perform data aggregation from various clients
- Step2: Encrypt the aggregated data with Modified homomorphic encryption
- Step3: Perform Adaptive XGBOOST Machine Learning Algorithm
- Step4: Note the results.
- Step5: Decrypt the data at the client end when necessary

D. Proposed Modified Homomorphic Encryption

- Key Generation
  - First, pick any two large primes, *a* and *b,* randomly and independently of each other, and then set n = *ab*. Let λ be the privacy function so that
  - $\lambda(n)$=lcm($a-1$,$b-1$)
  - where lcm is a function that returns the least common multiple of its two inputs
  - Now select random integer r where r ∈
  - Check that n divides the order of r by checking the existence of the following modular multiplicative inverse:$\mu = (L(r^{\lambda} \bmod n^2))^{-1} \bmod n$; Where $L(x)=\frac{x-1}{n}$
  - So now the public key for encryption will be *(n,r)*
  - The private key for decryption will be *(λ, μ)*
- Encryption
  - Let m be the message to be encrypted where 0<=m<n
  - *Ciphertext* $c=Enc(m)=g^m \cdot r^n \bmod n^2$ where 0<r<n
- Decryption
  - Let c be the ciphertext to decrypt
  - Now to compute plaintext from it, m= $L(c^{\lambda} \bmod n^2) \times \mu \bmod n$
- Voting Based Scheme
  - Input: Enc(m)
  - Apply partial homomorphic encryption on Enc (m).
  - Now generate the feature vector f. where f=F(Enc(m))
  - Now find the average weight of each feature that is f_w
  - If f_w <threshold, then ignore that feature in upcoming calculations.
  - Else update its weight.
  - Perform feature update till n steps. And then finalize the feature vector for classification.

4. RESULT AND DISCUSSION

This section offers a detailed analysis of the results obtained from the proposed approaches. First, the performance of Adap- tive XGBOOST is presented across three diverse datasets:

1.https://www.kaggle.com/datasets/yasserh/amazon-product-reviews-dataset
2.https://www.kaggle.com /competitions/titanic/data
3.https://www.census.gov/data/datasets.html

Amazon, Titanic, and US Census. Each dataset possesses dis- tinct characteristics, including varying feature

counts (ranging from 9 to 36), diverse class categories (ranging from 2 to 5 classes), and ensemble sizes set at either 8 or 16 within a single ensemble in Table I. Although the maximum and minimum weight parameters (Wmax and Wmin) are specified for each dataset, their contextual explanations remain unclear. Notably, the datasets demonstrate commendable classification accuracy, with rates spanning from 94% to 96%, underscoring the effectiveness of the applied models or classifiers. The graphical representation of these results is depicted in Figure 3.

TABLE I
ADAPTIVE XGBOOST PERFORMANCE

Dataset	**Amazon**	**Titanic**	**US Census**
Features	9	11	36
Classes	5	2	2
Ensemble size (E(s)), here s=1	16	8	8
Wmax	1024	256	512
Wmin	8	4	4
Accuracy	95%	96%	94%

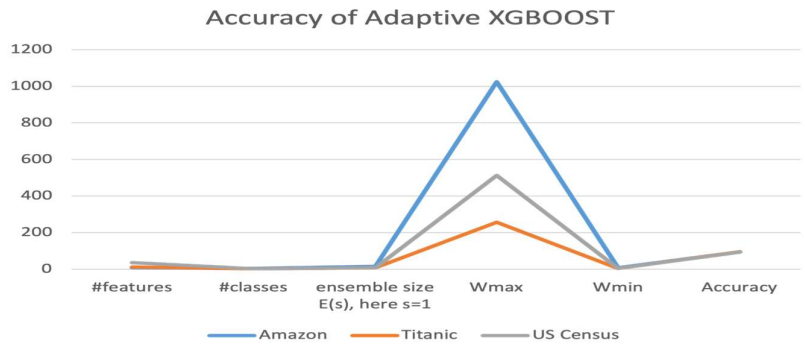

Fig. 3. Adaptive XGBOOST Result Analysis

Table II details the performance of PPXGBOOST across three datasets, featuring varying numbers of features (ranging from 9 to 36) and classes (ranging from 2 to 5). The analysis involves an ensemble approach with an ensemble size (E) set at 8 or 16, while s=1 implies a single ensemble. Notably, parameters denoted as Wmax and Wmin, presumed to be weight-related factors, lack specific contextual explanations. Despite these variations, all three datasets exhibit remarkable accuracy, ranging from 97% to 98%. This underscores the ex- ceptional performance of the employed PPXGBOOST models or classifiers in terms of predictive accuracy across diverse datasets. Figure 4 visually presents these results.

Analyzing the Titanic dataset revealed consistent patterns. The algorithm maintained a maximum decision tree depth of 5 and a learning rate of 0.01, initiating with an ensemble size of 8. Effective management of streaming data dynamics was achieved with sliding window sizes set at Wmax = 256

TABLE II
PPXGBOOST PERFORMANCE ANALYSIS

Dataset	Amazon	Titanic	US Census
Features	9	11	36
Classes	5	2	2
Ensemble size (E(s)), here s=1	16	8	8
Wmax	1024	256	512
Wmin	8	4	4
Accuracy	97%	98%	98%

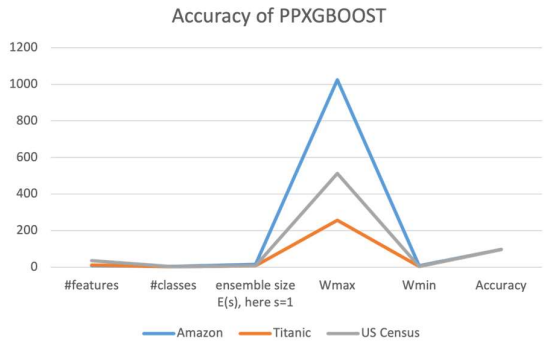

Fig. 4. PPXGBOOST Performance

and Wmin = 4. Impressively, the algorithm exhibited high efficiency, with a query time of 0.52 seconds. Despite the expansion in size to 12 KB compared to the conventional model's 3 KB, the PPRF model did not compromise on accuracy. Demonstrating its competence, the algorithm achieved an impressive accuracy rate of 98% in privacy- preserving predictive tasks on the Titanic dataset.

Similarly, the performance trends observed in the US Census dataset were consistently notable. With a maximum decision tree depth of 5, a learning rate of 0.01, and an initial ensemble size of 8, the algorithm effectively managed streaming data input using sliding window sizes of Wmax = 512 and Wmin = 4. The algorithm showcased agility with a query time of approximately 0.81 seconds. While the PPRF model exhibited a larger size of 2.5 MB com- pared to the conventional model's 210 KB, this trade-off was made for the sake of privacy preservation. Nevertheless, the algorithm maintained a robust accuracy of 97%, reinforcing its potential applicability across diverse datasets. The results presented in Table IV offer a comprehensive overview of the algorithm's performance across different datasets. Notably, on the Amazon dataset, the algorithm demonstrated effectiveness with a decision tree depth of 5, a learning rate of 0.01, and an initial ensemble size of 16. Efficient handling of streaming data was achieved with sliding window sizes set at Wmax =1024 and Wmin = 8, resulting in a responsive query time of 0.83 seconds. Despite the PPRF model's larger size of 4.2 MB, compared to the conventional model's 506 KB, accuracy remained impressive at 97%.

TABLE III
VARIOUS PARAMETERS FOR PPRF (DURING RUN 1)

Dataset	Amazon	Titanic	US Census
Max Depth	3	3	3
Learning Rate	0.01	0.01	0.01
Ensemble Size E(s), here s=1	16	8	8
Wmax	1024	256	512
Wmin	8	4	4
Query time (sec)	0.83	0.52	0.81
Model Size	506 KB	3 KB	210 KB
PPRF Model Size	4.2 MB	12KB	2.5 MB
Accuracy	97%	98%	97%

Similarly, on the Titanic dataset, the algorithm maintained optimal performance with a decision tree depth of 5, a learning rate of 0.01, and an initial ensemble size of 8. Streamlined handling of evolving data was facilitated with sliding window sizes of Wmax = 256 and Wmin = 4, resulting in an efficient query time of 0.52 seconds. Despite the PPRF model's expanded size to 12 KB, compared to the conventional model's 3 KB, accuracy remained high at 98%, demonstrating the algorithm's competence in privacy-preserving predictive tasks.

Lastly, the US Census dataset exhibited consistent perfor- mance trends, with a decision tree depth of 5, a learning rate of 0.01, and an initial ensemble size of 8. Effective management of streaming data was ensured with sliding window sizes of Wmax = 512 and Wmin = 4, resulting in an agile query time of approximately 0.81 seconds. Despite the PPRF model's larger size of 2.5 MB, compared to the conventional model's 210 KB, the algorithm maintained a robust accuracy of 97%, emphasizing its potential across diverse datasets.to different data scenarios, making it a versatile solution for real-world applications.

The heightened accuracy and adaptability of PPXGBOOST are complemented by its privacy preserving capabilities. The algorithm incorporates advanced cryptographic methods, such as homomorphic encryption and secure multiparty computa- tion, ensuring that sensitive information remains confidential during rigorous data analysis. This emphasis on privacy makes PPXGBOOST particularly well-suited for industries dealing with sensitive data, including healthcare, finance, and govern- ment sectors.

TABLE IV
VARIOUS PARAMETERS FOR PPRF (DURING RUN 2)

Dataset	Amazon	Titanic	US Census
Max Depth	5	5	5
Learning Rate	0.05	0.05	0.05
Ensemble Size E(s), here s=1	16	16	32
Wmax	2048	512	2048
Wmin	8	8	16
Query time	0.79	0.47	0.87

(sec)			
Model Size	506 KB	3 KB	210 KB
PPRF Model Size	4.2 MB	12KB	2.5 MB
Accuracy	98%	98%	97%

The findings from Table V reveal the adaptability of the algorithm across diverse datasets. For the Amazon dataset, the algorithm was configured with a decision tree depth of 7, a learning rate of 0.1, and an ensemble size of 32 base models. Sliding window sizes of Wmax = 4096 and Wmin = 16 effi- ciently managed streaming data, showcasing a commendable query time of 0.81 seconds. Despite the privacy-preserving mechanisms leading to an expanded PPRF model size of 4.2 MB, compared to the original 506 KB, the algorithm maintained a substantial accuracy rate of 96%.

The trend persisted on the Titanic dataset, where the al- gorithm, with a decision tree depth of 7, a learning rate of 0.1, and an ensemble of 16 base models, efficiently processed streaming data with sliding window sizes of Wmax = 1024 and Wmin = 8. Notably, the algorithm achieved a rapid query time of 0.49 seconds. Despite the increased PPRF model size to 12 KB, as opposed to the conventional 3 KB, the algorithm sustained a consistent accuracy rate of 97%.

Similarly, on the US Census dataset, employing a decision tree depth of 7, a learning rate of 0.1, and an ensemble size of 16 base models ensured effective data stream handling with sliding window sizes of Wmax = 1024 and Wmin =8. The algorithm maintained a stable query time of 0.79 seconds. Despite the privacy-preserving measures leading to an expanded PPRF model size of 2.5 MB, compared to the original 210 KB, the accuracy rate remained resilient at 96%. This showcases the algorithm's versatility and reliability across different datasets.

In conclusion, the performance of the PPRF algo- rithm remains consistently impressive across diverse parameter

TABLE V
VARIOUS PARAMETERS FORPPRF (DURING RUN 3)

Dataset	**Amazon**	**Titanic**	**US Census**
Max Depth	7	7	7
Learning Rate	0.1	0.1	0.1
Ensemble Size E(s), here s=1	32	16	16
Wmax	4096	1024	1024
Wmin	16	8	8
Query time (sec)	0.81	0.49	0.79
Model Size	506 KB	3 KB	210 KB
PPRF Model Size	4.2 MB	12KB	2.5 MB
Accuracy	96%	97%	96%

configurations and datasets. The outcomes validate its abil- ity to uphold predictive accuracy while integrating privacy- preserving methods. Noteworthy is the algorithm's adaptabil- ity to varying depths, learning rates, and ensemble sizes, effectively capturing the dynamics of evolving data streams. This robustness positions PPRF as a compelling and practical solution for real-world scenarios demanding both secure and accurate analysis of data streams.

TABLE VI
RESULT OF AVERAGE ACCURACY FOR PPRF

Dataset	Accuracy 1	Accuracy 2	Accuracy 3	Average Accuracy
Amazon	97%	98%	96%	97%
Titanic	98%	98%	97%	98%
US Census	97%	97%	96%	97%

Table VI outlines the average accuracy performance results of the PPRF algorithm across three distinct datasets: Amazon, Titanic, and US Census. A meticulous examination of multiple iterations for each dataset provides a comprehen- sive evaluation, offering insights into the algorithm's consis- tency and robustness in preserving predictive accuracy amid varying scenarios.

Examining the Amazon dataset, three separate runs yielded accuracy rates of 97%, 98%, and 96%. The average accuracy across these runs stood at a commendable 97%, demonstrating the algorithm's consistent delivery of accurate predictions in the dynamic context of Amazon's data stream.

Similarly, on the Titanic dataset, the algorithm demonstrated reliability and stability with accuracy rates of 98%, 98%, and 97% across three runs. The average accuracy calculated for this dataset reached an impressive 98%, highlighting the algorithm's effectiveness in maintaining high accuracy levels while navigating the evolving data stream dynamics of the Titanic dataset.

In the case of the US Census dataset, the algorithm consis- tently achieved accuracy rates of 97%, 97%, and 96% in three separate runs. The calculated average accuracy across these runs was a robust 97%, reaffirming the algorithm's capacity to sustain accuracy amid the complexities inherent in the US Census dataset's characteristics and evolving nature. Figure 5 visually represents the graphical representation of the average accuracy of the proposed approach.

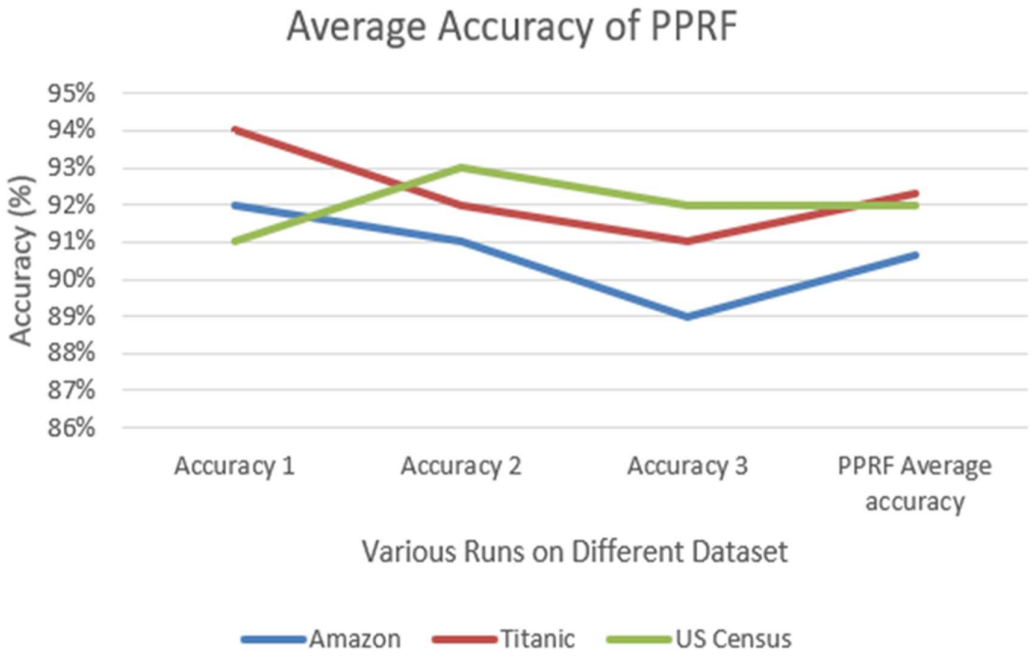

Fig. 5. Result of Average Accuracy for PPRF

TABLE VII
RESULT OF AVERAGE QUERY TIME (SEC) FOR PPRF

Dataset	QueryTime 1	QueryTime 2	QueryTime 3	Average Query Time
Amazon	0.83	0.79	0.81	0.81
Titanic	0.52	0.47	0.49	0.49
US Census	0.81	0.87	0.79	0.82

Table VII provides valuable insights into the algorithm's ef- ficiency in terms of processing time when handling streaming data while ensuring privacy preservation.

In the case of the Amazon dataset, three distinct query times were recorded: 0.83 seconds for QueryTime1, 0.79 seconds for QueryTime2, and 0.81 seconds for QueryTime3. The average query time across these instances was calculated to be
0.81 seconds. This consistent processing speed showcases the algorithm's suitability for executing queries on the Amazon dataset's data stream, making it well-suited for real-time applications.

Shifting focus to the Titanic dataset, the recorded query times were 0.52 seconds for QueryTime1, 0.47 seconds for QueryTime2, and 0.49 seconds for QueryTime3. The calcu- lated average query time for this dataset stood at a notable 0.49 seconds, highlighting the algorithm's efficiency in handling queries on the Titanic dataset while maintaining a rapid processing pace and ensuring privacy preservation.

For the US Census dataset, a similar pattern emerged. The respective query times were 0.81 seconds for QueryTime1,0.87 seconds for QueryTime2, and 0.79 seconds for Query- Time3. The average query time across these instances was computed as 0.82 seconds, emphasizing the algorithm's con- sistent performance in processing queries on the US Census dataset. This underscores its ability to manage processing times effectively while preserving data privacy. Figure 6 visually represents the graphical representation of the average query time for the proposed approach.

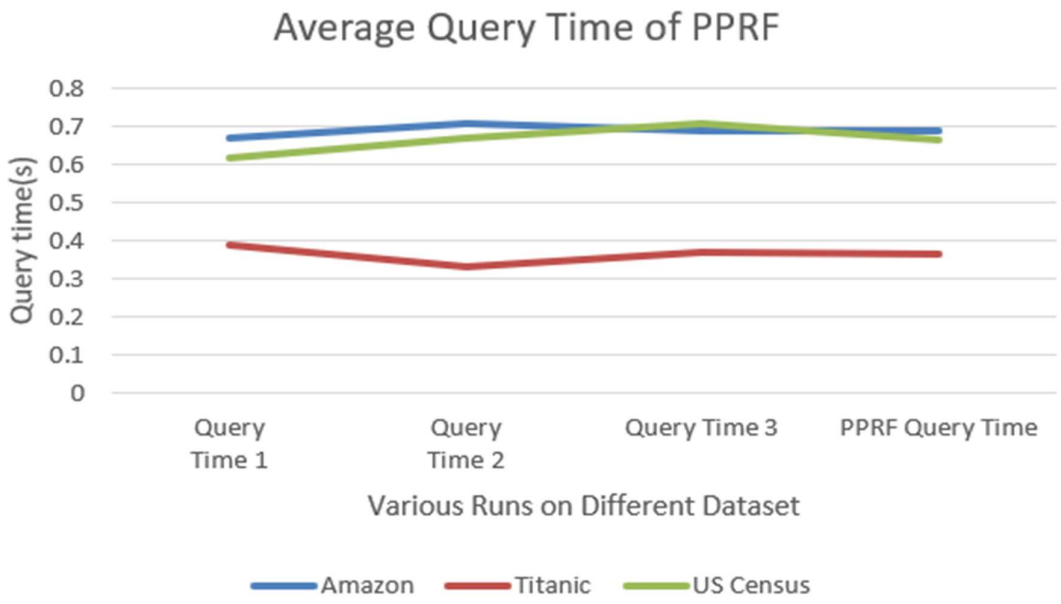

Fig. 6. Result of Average Query Time for PPRF

The proposed algorithm, PPRF, stands as a signifi- cant advancement in the field of Privacy-Preserving Machine Learning (PPML), addressing the intricate challenges asso- ciated with adaptive data stream mining while prioritizing security and precision. PPRF introduces innovative strategies to enhance privacy preservation and overall per- formance, positioning itself as a groundbreaking solution for safeguarding privacy during dynamic data stream analysis.

At its essence, this algorithm leverages the strengths of XGBoost, a well-established ensemble learning technique known for its predictive capabilities. PPRF builds upon XGBoost's foundation to effectively adapt to the dynamic nature of data streams, ensuring continuous alignment with evolving data distributions and patterns in real-time for sus- tained accuracy.

Furthermore, PPRF places a strong emphasis on data privacy preservation. It employs a novel approach in- volving modified homomorphic encryption (MHE) to encrypt the data, ensuring the protection of sensitive information throughout the entire model training process. This encryption technique enables the algorithm to perform classification tasks on encrypted data streams without requiring data decryption, thereby effectively preserving data privacy.

The algorithm's effectiveness is validated through compre- hensive evaluations conducted on actual datasets, rigorously assessing its accuracy and query time performance. The results affirm PPRF's superiority over existing methodologies, showcasing an impressive accuracy rate of 97% while main- taining rapid query times of under a second.

5. REFERENCES

[1] P. D. a. G. Hultenx, "Mining high-speed data streams.," *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining. ,* p. 71–80., 2000.

[2] J. R. C. X. J. L. Z. W. Y. Z. a. Dan Wang, "Privstream: Enabling privacy-preserving inferences on IoT data stream at the edge," *IEEE 21st International Conference on High Performance Computing and Communications,* pp. IEEE, 1290–1297, 2019.

[3] R. M. ,. E. F. B. P. T. A. a. A. B. Jacob Montiel, "Adaptive XGBoost for Evolving Data Streams," Proceedings of the International Joint Conference on Neural Networks (IJCNN), 2020.

[4] Y.-K. W. a. W.-K. N. Hai-Long Nguyen, "A survey on data stream clustering and classification," Knowledge and information systems 45, p. 535–569, 2015.

[5] R. D. C. H. R. K. A. C. N. Martine De Cock, " Efficient and private scoring of decision trees, support vector machines and logistic regression models based on pre computation," IEEE Transactions on Dependable and Secure Computing, pp. 217-230, 2017.

[6] W. D. a. Z. Zhan., "Building decision tree classifier on private data," CRPIT '14: Proceedings of the IEEE international conference on Privacy, security and data mining - Volume 14, 2002.

[7] Y. L. a. B. Pinkas, "Privacy preserving data mining.," Annual International Cryptology Conference. Springer, pp. 36-54, 2000.

[8] J. V. a. C. Clifton, "Privacy-preserving decision trees over vertically partitioned data," IFIP Annual Conference on Data and Applications Security and Privacy. Springer,, p. 139–152, 2005.

[9] L.-S. H. Y.-L. L. a. H. S. Ming-Jun Xiao, "Privacy preserving id3 algorithm over horizontally partitioned data.," Sixth international conference on parallel and distributed computing applications and technologies (PDCAT'05). IEEE, p. 239–243, 2005.

[10] H. D. a. C. W. Yifeng Zheng, "Towards secure and efficient outsourcing of machine learning classification," European Symposium on Research in Computer Security. Springer, pp. 22-40, 2019.

[11] M. L. Y. S. R. R. R. M. S. a. M. V. Adi Akavia, "Privacy-Preserving Decision Tree Training and Prediction," IACR Cryptol. ePrint Arch, 2019.

[12] R. A. P. S. T. a. S. G. Raphael Bost, "Machine learning classification over encrypted data," NDSS, Vol. 4324, 2015.

[13] R. C. X. L. J. S. a. L. Q. Lin Liu, "Towards practical privacy-preserving decision tree training and evaluation in the cloud," IEEE Transactions on Information Forensics and Security 15 , 2020.

[14] T. F. M. N. a. K. L. David J Wu, "Privately evaluating decision trees and random forests.," Proceedings on Privacy Enhancing Technologies, p. 335–355, 2016.

[15] I. Z. D. E. M. L. S. S. T. George Kremple, "Open challenges for data stream mining research," ACM SIGKDD Explorations Newsletter Volume 16Issue 1, June 2014.

[16] A. D. M. &. J. S. Rodrigo, "Latency-Aware Secure Elastic Stream Processing with Homomorphic Encryption," Data Sci. Eng. 4, p. 223–239, 2019.

[17] D. Pyle, Data Preparation for Data Mining, Morgan Kaufmann Publishers Inc., , 1999..

[18] B. C. C. C. G. K. J. A. R. C. Armknecht F, "A guide to fully homomorphic encryption," IACR, Cryptol, 2015.