Penggunaan Shamir's Secret Sharing (Secure Aggregation) dalam Federated Learning

David Karel Halomoan - 13520154 (*Author*)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 13520154@std.stei.itb.ac.id

***Abstract*—Makalah ini membahas penggunaan Shamir's Secret Sharing dan Secure Aggregation dalam Federated Learning untuk meningkatkan privasi dan keamanan dalam proses agregasi hasil pelatihan yang terdistribusi. Dengan menggunakan Shamir's Secret Sharing, model dan parameter sensitif dibagi menjadi beberapa bagian yang tersebar di perangkat pengguna, menjaga privasi data dan menghindari pihak yang tidak berwenang mendapatkan akses penuh. Secure Aggregation memastikan agregasi dilakukan secara aman tanpa mengorbankan privasi, dengan menerapkan teknik kriptografi dan privasi pada hasil pelatihan yang dikumpulkan. Makalah ini menyajikan konsep dasar, penerapan, dan manfaat dari kedua metode ini dalam konteks Federated Learning, memberikan panduan penting bagi praktisi dan peneliti yang tertarik dalam menjaga keamanan dan privasi dalam sistem terdistribusi.**

Keywords—Federated Learning, Shamir's Secret Sharing, Secure Aggregation, Privasi Data, Keamanan, Rahasia, Agregasi, Pembelajaran Mesin Terdistribusi

I. PENDAHULUAN

Federated Learning adalah paradigma pembelajaran mesin terdistribusi yang memungkinkan model pembelajaran mesin untuk dilatih secara terdistribusi di berbagai perangkat yang ada pada jaringan terdistribusi tanpa mentransfer data yang sensitif ke pusat data sentral. Konsep ini memungkinkan pengguna atau perangkat yang memiliki data untuk tetap menjaga privasi mereka sementara memberikan kontribusi pada pelatihan model yang global dan memperoleh manfaat dari model tersebut. Federated Learning mengatasi beberapa tantangan utama dalam pembelajaran mesin sentralistik, seperti latensi jaringan, kebutuhan akan *bandwidth* tinggi, dan masalah privasi.

Penggunaan Federated Learning sangat penting dalam konteks saat ini di mana data pribadi yang sensitif semakin banyak terkumpul di perangkat pintar, seperti ponsel cerdas, sensor Internet of Things (IoT), dan perangkat *wearable*. Dengan mengadopsi Federated Learning, kita dapat mempertahankan privasi pengguna sambil memanfaatkan pengetahuan yang diperoleh dari data yang tersebar luas untuk meningkatkan kualitas model pembelajaran mesin. Paradigma ini juga memungkinkan organisasi atau institusi untuk berkolaborasi dalam membangun model yang lebih baik tanpa mengorbankan kerahasiaan data.

Namun, meskipun Federated Learning menawarkan manfaat yang signifikan dalam hal privasi, ada beberapa ancaman yang perlu diperhatikan. Salah satu ancaman utama adalah risiko kebocoran informasi pribadi selama proses pelatihan model di perangkat yang terdistribusi. Karena data tetap berada di perangkat pengguna, ada potensi untuk penyerangan yang bertujuan untuk memperoleh informasi sensitif tersebut. Selain itu, metode agregasi yang digunakan untuk menggabungkan pembaruan model dari perangkat yang berbeda juga dapat memberikan celah keamanan jika tidak diimplementasikan dengan benar.

Untuk mengatasi masalah keamanan dan privasi dalam Federated Learning, dalam makalah ini, kami memperkenalkan penggunaan *secure* aggregation yang menggunakan Shamir's Secret Sharing. Secure Aggregation adalah teknik yang memungkinkan agregasi data yang aman di dalam Federated Learning. Dalam konteks Federated Learning, setiap perangkat pengguna melakukan pelatihan model lokal dengan menggunakan data yang dimilikinya. Namun, untuk membangun model global yang baik, perlu dilakukan agregasi hasil pelatihan dari semua perangkat. Secure Aggregation memastikan bahwa agregasi dilakukan secara rahasia, sehingga data sensitif tidak terungkap dalam proses tersebut.

Shamir's Secret Sharing adalah sebuah metode kriptografi yang dikembangkan oleh Adi Shamir pada tahun 1979. Metode ini memungkinkan pemecahan suatu rahasia menjadi beberapa bagian yang disebarkan ke berbagai pihak yang dipercayai, dengan membutuhkan sejumlah tertentu dari bagian-bagian tersebut untuk mengembalikan rahasia asli. Dalam konteks Federated Learning, metode ini dapat digunakan untuk membagi model dan parameter yang sensitif menjadi beberapa bagian yang tersebar di perangkat yang berpartisipasi.

II. LANDASAN TEORI

A. Sistem Terdistribusi

Sistem Terdistribusi adalah kumpulan komputer atau node yang bekerja sama dan saling berkomunikasi untuk mencapai

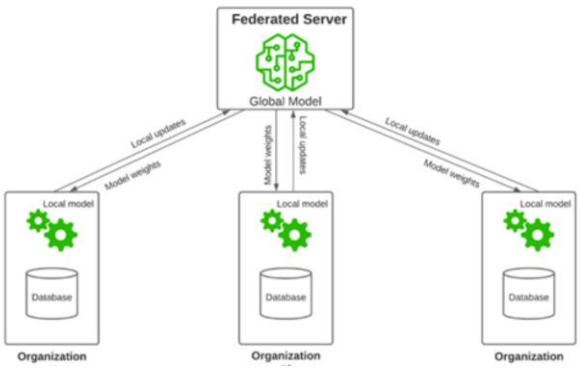
tujuan bersama. Konsep dasar dari sistem terdistribusi adalah memanfaatkan sumber daya dan kapabilitas dari beberapa komputer atau node untuk meningkatkan kinerja, skalabilitas, dan kehandalan sistem secara keseluruhan. Sistem Terdistribusi dapat digunakan dalam berbagai aplikasi, termasuk basis data terdistribusi, komputasi terdistribusi, dan layanan jaringan terdistribusi.

Arsitektur terdistribusi merujuk pada desain sistem terdistribusi yang melibatkan struktur organisasi dan pola komunikasi antara komputer atau node. Beberapa arsitektur terdistribusi yang umum digunakan termasuk arsitektur client-server, arsitektur peer-to-peer, dan arsitektur terdistribusi yang berbasis pada middleware. Arsitektur ini mempengaruhi cara komputasi dan komunikasi dilakukan dalam sistem terdistribusi.

Federated Learning adalah sebuah sistem terdistribusi dengan arsitektur *client server*. Federated Learning adalah pendekatan terdesentralisasi untuk pembelajaran mesin di mana proses pelatihan berlangsung di beberapa perangkat terdistribusi atau perangkat tepi (*edge*), bukan di server atau pusat data terpusat.

B. *Federated Learning*

Federated Learning adalah paradigma pembelajaran mesin terdistribusi yang memungkinkan pelatihan model terjadi di perangkat yang tersebar luas tanpa mentransfer data yang sensitif ke pusat data sentral. Konsep ini bertujuan untuk menjaga privasi pengguna sambil memanfaatkan pengetahuan yang diperoleh dari data yang tersebar untuk membangun model pembelajaran mesin yang lebih baik. Dalam Federated Learning, pelatihan model terjadi secara terdistribusi di perangkat pengguna, dan hanya pembaruan parameter yang dikirimkan ke pusat data untuk dilakukan agregasi. Dengan demikian, Federated Learning mengatasi tantangan privasi dan keamanan yang terkait dengan mentransfer data ke pusat data sentral.



Gambar 2.1 Ilustrasi Federated Learning
Sumber:
https://gemmo.ai/federated-learning

Pada tahap agregasi dalam Federated Learning, hasil pelatihan dari berbagai perangkat pengguna dikumpulkan untuk membangun model global yang lebih baik. Namun, karena data pelatihan tersebar di perangkat pengguna, perlu adanya teknik agregasi yang memastikan kerahasiaan dan keamanan data. Beberapa teknik yang umum digunakan dalam Federated Learning adalah Secure Aggregation, Homomorphic Encryption, dan Differential Privacy. Secure Aggregation memungkinkan agregasi yang aman tanpa mengungkapkan informasi sensitif yang dimiliki oleh perangkat pengguna. Homomorphic Encryption memungkinkan komputasi yang aman pada data yang dienkripsi, sedangkan Differential Privacy memastikan bahwa hasil agregasi tidak mengungkapkan informasi pribadi tentang individu yang berpartisipasi.

Terdapat berbagai algoritma dan teknik yang digunakan dalam Federated Learning untuk melakukan pelatihan model di perangkat pengguna. Beberapa di antaranya termasuk Federated Averaging, Federated Proximal, dan Federated Stochastic Gradient Descent (SGD). Federated Averaging adalah algoritma yang paling umum digunakan dalam Federated Learning. Algoritma ini melibatkan perangkat pengguna yang melakukan pelatihan model lokal, mengirimkan pembaruan parameter ke pusat data, dan melakukan agregasi dengan memperbarui model global berdasarkan hasil agregasi. Algoritma Federated Proximal menggunakan pendekatan optimasi secara iteratif untuk memperbarui parameter model di perangkat pengguna dan melakukan agregasi. Federated SGD merupakan variasi dari algoritma Stochastic Gradient Descent yang dikustomisasi untuk Federated Learning. Algoritma yang dibahas pada makalah ini dan diimplementasikan adalah Federated Averaging.

Server executes:
initialize w_0
for each round $t = 1, 2, \cdots$ do
$m \leftarrow \max(C \cdot K, 1)$
$S_t \leftarrow$ (random set of m clients)
for each client $k \in S_t$ do (in parallel)
$w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$
$w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$
where $0 \leq C \leq 1$ is a hyperparameter, K is the total number of clients
ClientUpdate(k, w):
$\mathcal{B} \leftarrow$ (split P_k into batches of size B)
for each local epoch i from 1 to E do
for batch $b \in \mathcal{B}$ do
$w \leftarrow w - \eta \nabla l(w; b)$
return w to server
where P_k is all the data on client k, η is some arbitrary learning rate, and $\nabla l(w; b)$ is the average gradient of the loss function when evaluated on batch b

Figure 2.2 Pseudocode dari Algoritma Federated Averaging
Sumber:
https://www.distributedgenomics.ca/posts/federated-learning-candig/

Pada makalah ini, algoritma pembelajaran mesin yang digunakan adalah *logistic regression*. *Update* yang diberikan *client* ke *server* pada algoritma *federated learning* juga disimplifikasi, yaitu perbedaan koefisien dan intersep dari model *logistic regression* pada *client* dan *server*.

C. *Logistic Regression*

Logistic Regression adalah metode klasifikasi yang digunakan untuk memprediksi probabilitas kejadian suatu peristiwa dengan menghubungkan variabel input (variabel independen) dengan variabel output (variabel dependen) yang bersifat biner atau kategorikal. Metode ini mengasumsikan hubungan logistik antara variabel input dan output, dan

menghasilkan prediksi dalam bentuk probabilitas antara 0 dan 1.

Dalam Logistic Regression, fungsi logistik atau sigmoid digunakan untuk memodelkan hubungan antara variabel input dan output. Fungsi sigmoid memiliki bentuk matematika sebagai berikut:

$$g(z) = 1 / (1 + e^{(-z)})$$

Di mana z adalah kombinasi linear dari variabel input dan parameter yang akan diestimasi.

Fungsi sigmoid memetakan nilai input ke dalam rentang probabilitas antara 0 dan 1. Nilai output yang mendekati 1 menunjukkan probabilitas yang tinggi untuk kelas positif, sedangkan nilai output yang mendekati 0 menunjukkan probabilitas yang tinggi untuk kelas negatif.

D. Shamir's Secret Sharing (Secure Aggregation)

Shamir's Secret Sharing memanfaatkan ide dari persoalan interpolasi. Dari persoalan ini didapatkan untuk membentuk polynomial

$$y = a_0 + a_1x + a_2x^2 + \ldots + a_nx_n$$

diperlukan $n + 1$ titik. Nilai a_0 sampai a_n dapat diperoleh dengan mebuat $n + 1$ buah persamaan lanjar (linear). Persamaan-persamaan lanjar ini lalu diselesaikan. Salah satu cara menyelesaikan persamaan-persamaan lanjar ini adalah dengan metode Gauss.

Shamir's Secret Sharing juga menggunakan skema ambang (*threshold schemes*). Skema ambang adalah metode pembagian *secret S* kepada *n* partisipan sedemikian sehingga sembarang himpunan bagian yang terdiri dari *t* partisipan dapat merekonstruksi *S*, tetapi jika kurang dari *t* maka *S* tidak dapat direkonstruksi.

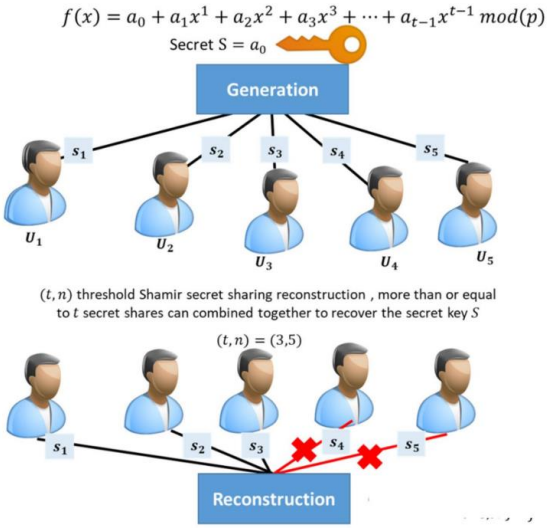
Algoritma pembagian *share* pada skema ambang adalah:

1. Pilih bilangan prima *p*, yang harus lebih besar dari semua kemungkinan nilai secret *S* dan juga lebih besar dari jumlah *n* partisipan. Semua komputasi dilakukan dalam modulus *p*.
2. Pilih $t - 1$ buah bilangan bulat acak dalam modulus *p*, misalkan $a_1, a_2, \ldots, a_{t-1}$ dan nyatakan polinomial:

   $$f(x) \equiv S + a_1x + a_2x^2 + \ldots + a_{t-1}x^{t-1} \pmod p$$

   sedemikian sehingga $f(0) \equiv S \pmod p$.
3. Untuk *n* partisipan, kita pilih integer berbeda, $x_1, x_2, \ldots, x_n \pmod p$ dan setiap orang memperoleh share (x_i, y_i) yang dalam hal ini

   $$y \equiv f(x_i) \pmod p$$



Gambar 2.3 Ilustrasi Pembagian Share pada Skema Ambang
Sumber:
https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2022-2023/36-Skema-Pembagian-Data-Rahasia-2023.pdf

Shamir Secure Aggregation menggunakan konsep dari *Shamir's Secret Sharing* yang menggunakan ide dari persoalan interpolasi. Dengan menggunakan teknik ini, pihak pengagregasi dapat menghitung penjumlahan nilai *secret value* dari semua pihak tanpa perlu mengetahui nilai *secret value* dari masing-masing pihak. Tahapan dalam *Shamir's Secure* Aggregation adalah:

1. Misal terdapat *n* pihak dan jumlah dari nilai rahasia (*secret value*, dilambangkan dengan *Sec_Val*) dari setiap pihak adalah 1. Ditentukan juga *t*, yaitu jumlah minimal pihak yang harus berpartisipasi untuk melakukan agregasi. Nilai *t* harus lebih kecil atau sama dengan *n*. Setiap pihak juga membangkitkan koefisien rahasia (*secret coefficient*) yang dilambangkan dengan c_1 hingga c_{k-1} sejumlah $k - 1$ untuk masing-masing pihak. Setiap pihak juga membangkitkan semuah nilai publik yang dilambangkan dengan *Pub_Val*.
2. Setiap pihak membentuk nilai R_y^x, x adalah sumber nilai R dikirim dan y adalah tujuan nilai R dikirim. Nilai R dihitung dengan menggunakan rumus:

   $$R_y^x = c_{k-1}.(Pub\_Val_y)^{k-1} + \ldots + c_1.(Pub\_Val_y)^1 + Sec\_Val_x$$

   Setiap pihak menghitung dan mengirimkan nilai R untuk dikirm ke semua pihak lainnya dan juga menghitung nilai R jika pihak tersebut mengirim R ke dirinya sendiri.
3. Semua pihak lalu menghitung nilai *intermediate result* (dilambangkan dengan *Inter_Res*) masing-masing dengan menjumlahkan semua nilai R yang diterima dan nilai R yang dikirm ke dirinya sendiri. Berikut ilustrasi hasil perhitungan *intermediate result* untuk sistem dengan 4 pihak:

$$
\begin{aligned}
Intr\_Res^{P_1} &= R_{P_1}^{P_1} + R_{P_1}^{P_2} + R_{P_1}^{P_3} + R_{P_1}^{P_4} \\
&= c_2^{P_1}.(PV_1)^2 + c_1^{P_1}.(PV_1)^1 + SV_1 \\
&+ c_2^{P_2}.(PV_1)^2 + c_1^{P_2}.(PV_1)^1 + SV_2 \\
&+ c_2^{P_3}.(PV_1)^2 + c_1^{P_3}.(PV_1)^1 + SV_3 \\
&+ c_2^{P_4}.(PV_1)^2 + c_1^{P_4}.(PV_1)^1 + SV_4 \\
&= (c_2^{P_1} + c_2^{P_2} + c_2^{P_3} + c_2^{P_4}).(PV_1)^2 + (c_1^{P_1} + c_1^{P_2} + c_1^{P_3} + c_1^{P_4}).(PV_1)^1 + (SV_1 + SV_2 + SV_3 + SV_4)
\end{aligned}
$$

Gambar 2.4 Ilustrasi Hasil Perhitungan untuk Sistem yang Terdiri dari 4 Pihak
Sumber:
https://www.youtube.com/watch?v=XKIqvFMmZJM

4. Setiap pihak lalu mengirim *intermediate result* ke sebuah pihak yang perlu melakukan agregasi. Pihak yang melakukan agregasi memiliki beberapa nilai yang diketahui dan tidak diketahui. Berikut ilustrasi nilai yang diketahui dan tidak diketahui oleh pihak pengaregasi untuk sistem dengan 4 pihak:

- Unknowns: $\begin{cases} c_2^{P_1} + c_2^{P_2} + c_2^{P_3} + c_2^{P_4} \\ c_1^{P_1} + c_1^{P_2} + c_1^{P_3} + c_1^{P_4} \\ SV_1 + SV_2 + SV_3 + SV_4 \end{cases}$
- Knowns: $\begin{cases} Intr\_Res^{P_1}, Intr\_Res^{P_2}, Intr\_Res^{P_3}, Intr\_Res^{P_4} \\ PV_1, PV_2, PV_3, PV_4 \end{cases}$

Gambar 2.5 Ilustrasi Nilai yang Diketahui dan Tidak Diketahui Oleh Pihak Pengaregasi dalam Sistem dengan 4 Pihak
Sumber:
https://www.youtube.com/watch?v=XKIqvFMmZJM

5. Pihak pengagregasi melakukan perhitungan agregasi *secret value* dengan nilai *intermediate result* dari semua pihak. Berikut ilustrasi perhitungan agregasi *secret value* untuk sistem dengan 4 pihak:

$$
\begin{cases}
Intr\_Res^{P_1} = (c_2^{P_1} + c_2^{P_2} + c_2^{P_3} + c_2^{P_4}).(PV_1)^2 + (c_1^{P_1} + c_1^{P_2} + c_1^{P_3} + c_1^{P_4}).(PV_1)^1 + (SV_1 + SV_2 + SV_3 + SV_4) \\
Intr\_Res^{P_2} = (c_2^{P_1} + c_2^{P_2} + c_2^{P_3} + c_2^{P_4}).(PV_2)^2 + (c_1^{P_1} + c_1^{P_2} + c_1^{P_3} + c_1^{P_4}).(PV_2)^1 + (SV_1 + SV_2 + SV_3 + SV_4) \\
Intr\_Res^{P_3} = (c_2^{P_1} + c_2^{P_2} + c_2^{P_3} + c_2^{P_4}).(PV_3)^2 + (c_1^{P_1} + c_1^{P_2} + c_1^{P_3} + c_1^{P_4}).(PV_3)^1 + (SV_1 + SV_2 + SV_3 + SV_4) \\
Intr\_Res^{P_4} = (c_2^{P_1} + c_2^{P_2} + c_2^{P_3} + c_2^{P_4}).(PV_4)^2 + (c_1^{P_1} + c_1^{P_2} + c_1^{P_3} + c_1^{P_4}).(PV_4)^1 + (SV_1 + SV_2 + SV_3 + SV_4)
\end{cases}
$$

Gambar 2.6 Ilustrasi Perhitungan Agregasi Secret Value untuk Sistem dengan 4 Pihak
Sumber:
https://www.youtube.com/watch?v=XKIqvFMmZJM

Persamaan-persamaan lanjar di atas dapat diselesaikan dengen berbagai metode, salah satunya adalah metode Gauss. Setelah persamaan-persamaan lanjar tersebut diselesaikan, akan ditemukan penjumlahan dari setiap nilai c_1 hingga c_{k-1} dari setiap pihak dan penjumlahan nilai *secret value* dari semua pihak. Nilai penjumlahan *secret value* inilah hasil yang dicari. Dari tahapan-tahapan yang ada, tidak ada pihak yang megirimkan *secret value*-nya secara langsung ke pihak lainnya sehingga keamanan terjamin.

III. Deskripsi Masalah

Federated Learning digunakan dalam berbagai hal, misalnya:

- *Privacy-Preserving Machine Learning*

  Federated Learning memungkinkan organisasi atau entitas untuk berkolaborasi dan melatih model machine learning tanpa harus berbagi data mentah. Hal ini membantu mengatasi masalah privasi dengan menjaga data tetap terlokalisasi sambil memungkinkan pembelajaran kolektif.

- *Healthcare*

  Di sektor kesehatan, *federated learning* memungkinkan kolaborasi antara beberapa rumah sakit, lembaga penelitian, atau penyedia layanan kesehatan sambil menjaga privasi data. Model dapat dilatih pada data terdistribusi untuk mengembangkan model prediktif untuk diagnosis penyakit, pengobatan personalisasi, atau rekomendasi pengobatan.

- *Internet of Things (IoT)*

  Federated Learning dapat diterapkan pada perangkat edge dalam ekosistem IoT. Perangkat seperti ponsel pintar, sensor, atau perangkat pintar lainnya dapat melatih model secara kolaboratif sambil menjaga data sensitif di perangkat tersebut. Pendekatan ini mengurangi kebutuhan untuk mentransmisikan sejumlah besar data ke server pusat.

- *Financial Services*

  Federated Learning berguna dalam institusi keuangan untuk deteksi penipuan, skor kredit, atau penilaian risiko. Beberapa bank atau organisasi keuangan dapat berkolaborasi untuk melatih model tanpa mengungkapkan data sensitif pelanggan.

Pada *Federated Learning*, data yang digunakan untuk pembelajaran (*training data*) tetap berada pada perangkat atau sistem masing-masing. Hal ini dilakukan agar data-data pribadi dan sensitif yang ada tidak tersebar keluar perangkat atau sistem data tersebut terjadi. Pada *federated learning*, *client* biasanya mengirimkan *update* (perubahan) yang didapat setelah dilakukan pelatihan model ke server pusat.

Walaupun hal ini menjaga privasi dari data yang ada pada pernagkat atau sistem, *update* yang dikirim dapat digunakan oleh pihak yang tidak bertanggung jawab untuk melakukan inferensi terhadap data yang ada pada perangkat atau sistem. Penyerang dapat mengetahui kecenderungan atau karakteristik dari data yang ada pada perangkat atau sistem dan menggunakan kecenderungan atau karakteristik ini untuk hal-hal yang tidak benar. Hal inilah yang menyebabkan pengiriman *update* ke server juga harus diamankan.

Salah satu cara yang dapat digunakan adalah dengan mengirimkan *update* dengan enkripsi homomorfik. Untuk beberapa algoritma seperti *federated averaging*, enkripsi homomorfik yang digunakan juga cukup parsial saja (misal aditif), tidak perlu penuh. Kelemahan penggunaan enkripsi homorfik adalah jika kunci dekripsi ketahuan atau tersebar, penyerang dapat mendekrisp *update* yang dikirim *client* dan

bahkan mendekripsi nilai hasil perhitungan agregasi dengan enkripsi homorfik.

Hal inilah yang menyebabkan penulis memutuskan untuk menggunakan *secure aggregation*. Dengan teknik ini, nilai agregasi yang diperlukan pada *federated averaging* hanya bisa didapatkan jika semua atau sejumlah besar *client* berpartisipasi dan nilai *update* individual dari *client* tidak dapat didapat, hanya nilai hasil agregasi yang dapat didapat.

IV. Analisis dan Implementasi

Pada algoritma yang dibuat oleh penulis, server pusat membangkitkan nilai *secret value* yang *random*. Nilai ini tentunya hanya diketahui oleh server tersebut. Hal ini menyebabkan *client* tidak dapat mengetahui hasil nilai agregasi dari *update* yang dikirim semua *client* Jika *client* melakukan agregasi *secret value* sesuai dengan langkah yang sudah dijelaskan, *client* akan mendapatkan nilai penjumlahan semua *secret value* dari *client* dan *secret* dari server. Hal ini menyebabkan nilai hasil agregasi yang didapat oleh *client* salah. Nilai agregasi dari semua update dari client yang benar hanya bisa didapatkan dengen mengurangi hasil agregasi keseluruhan dengan *secret value* dari server. Hal inilah yang menjamin keamanan dari agregasi semua *update* dari *client*. Hanya server pusat yang dapat dan berhak mengetahui hasil agregasi nilai *update* dari semua *client*.

Implementasi program dilakukan dengan bahasa Python. Algoritma pembelajaran mesin yang digunakan adalah *logistic regression* dengan pustaka sklearn. Dataset yang digunakan adalah dataset *breast cancer* yang terdapat pada sklearn. Dataset yang digunakan memiliki dua kelas (*binary class*). Implementasi algoritma yang dibuat mengasumsikan dataset dengan *binary class*. Dataset ini akan dibagi secara terpisah ke sejumlah *client.* Pada implementasi, data yang dibagikan pada setiap *client* bersifat *disjoint*. Implementasi algoritma juga mengasumsikan semua *node* atau pihak (termasuk server) wajib berpartisipasi dalam agregasi *secret value*. *Client* membagi dataset-nya msing-masing menjadi *traing data* dan *test data* (pada implementasi, 80% data digunakan sebagai *training data* dan 20% sebagai *test data*).

Pertama, server menginisialisasi model yang mengirimkan konfigurasi model tersebut ke semua *client*-nya. Server lalu memerintahkan *client* untuk melakukan *training* (pembelajaran atau pelatihan). Setelah setiap *client* telah melakukan *training*, setiap *client* mengitung perbedaan (selisih) koefisien dan intersep sesudah dan sebelum *training*. Setiap *client* lalu mengalikan setiap perbedaan tersebut dengan jumlah data yang digunakan untuk *training*. Nilai-nilai ini disebut sebagai *weighted updates*. *Client* lalu menetapkan jumlah data *training* dan *weighted updates* sebagai *secret value*-nya. Pada algoritma *logistic regression*, koefisien dapat bernilai banyak sehingga *secret value* dari *client* berbentuk *array*. Elemen pertama pada *array* tersebut adalah jumlah *training data*, elemen kedua pada *array* tersebut adalah *weighted update* dari intersep, dan sisanya adalah *weighted updates* dari koefisien-koeifisien yang ada. Setiap elemen pada *array* ini akan dijumlahkan dengan elemen dengan urutan sama dari *client* lainnya oleh *server* dengan *secure aggregation*. Dengan cara ini, setelah server melakukan *secure aggregation*, server akan mendapatkan *array* dengan elemen pertama penjumlahan data dari semua *client*, elemen kedua *weighted sum* dari semua intersep dari tiap *client*, dan sisanya *weighted sum* dari tiap koefisien dari tiap *client*. Server lalu membagi semua elemen pada *array* mulai dari elemen kedua dengan elemen pertama (penjumlahan data dari semua *client*) untuk mendapatkan *weighted average* dari *update* dari semua *client*. Server lalu menambahkan koefisien dan intersep-nya dengan *weighted average* tadi untuk melakukan *update* model. Server lalu mengirimkan konfigurasi model yang baru ke setiap *client*.

Dengan tahapan di atas, tidak ada satu *client* pun yang tahu jumlah data pada *client* lainnya atau jumlah data dari seluruh *client*. *Client* juga tidak mengetahui *update* dari *client* lain.

Berikut perbandingan hasil pembelajaran mesin dengan teknik biasa (*non-federated*) dan *federated learning* menggunakan *Shamir's Secure Aggregation*:

- Hasil dari teknik biasa (*non-federated*):

```
Accuracy = 0.956140350877193
```

Gambar 4.1 Hasil Akurasi Pembelajaran Mesin dengan Teknik Biasa
Sumber:
Dokumen pribadi penulis

- Hasil dari *federated learning* menggunakan *Shamir's Secure Aggregation*:

```
Training on each client:

Round 1
Client 1 training:
Local Accuracy = 0.9629629629629629
Client 2 training:
Local Accuracy = 0.9080459770114943

******************************
Evaluation on each client:
Client 1 evaluation:
Local Accuracy = 0.9259259259259259
Client 2 evaluation:
Local Accuracy = 0.9195402298850575
```

Gambar 4.2 Hasil Akurasi Pembelajaran Mesin dengan Teknik Federated Learning
Sumber:
Dokumen pribadi penulis

Dari hasil akurasi kedua teknik terlihat performa yang cukup baik pada algoritma *Federated Learning*. Akurasi saat *training* menunjukkan akurasi *client* terhadap *test data* pada masing-masing *client* setelah dilakukan *training*. Akurasi evaluasi menunjukkan akurasi *client* terhadap *test data* pada masing-masing *client* setelah server melakukan agregasi dari

hasil *update* setiap *client* dan melakukan *update* model ke setiap *client*. Dari sini juga terlihat dan terbukti Shamir's Secret Sharing dapat digunakan pada *Federated Learning* dengan teknik *Federated Averaging*. *Round* (ronde) pada implementasi hanya dapat diisi dengan 1. Jumlah ronde disediakan untuk mempermudah pengimplementasian *mini-batch* di masa depan.

V. Kesimpulan dan Saran

Privasi data merupakan salah satu persoalan penting yang makin diperlukan terutama dalam dunia dengan perkembangan digital yang melesat sekarang ini. Penggunaan *machine learning* (pembelajaran mesin) yang kian marak membuktikan perlunya solusi terhadap permasalahan privasi dalam *machine learning*, terutama dalam *machine learning* terdistribusi seperti *federated learning*. Dari makalah ini dapat disimpulkan *Shamir's Secure Aggregaion* dapat menjadi salah satu cara menghadapi persoalan ini. Algoritma ini juga mudah dimengerti dan memiliki efisiensi yang baik.

Untuk penelitian lebih lanjut, dapat diimplementasikan algoritma yang dapat meng-*handle* data dengan kelas lebih dari dua, mengimplementasikan *mini-batch*, atau heuristic dalam pengiriman dan pengaregasian *update*.

VI. Ucapan Terima Kasih

Penulis mengucapkan puji syukur kepada Tuhan Yang Maha Esa, karena atas kasih, rahmat, dan karunia-Nya sehingga penulis dapat menyelesaikan makalah ini. Penulis juga mengucapkan terima kasih kepada Bapak Rinaldi Munir, dosen mata kuliah Kriptografi IF4020 atas bimbingan dan pengajarannya selama ini. Penulis juga mengucapkan terima kasih kepada para penulis referensi makalah ini karena tanpa karya mereka, makalah ini tidak akan jadi.

GitHub Link

https://github.com/davidkarelh/Cryptography-in-Federated-Learning

Referensi

[1] https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2022-2023/36-Skema-Pembagian-Data-Rahasia-2023.pdf

[2] https://www.youtube.com/watch?v=XKIqvFMmZJM

Pernyataan

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 22 Mei 2023

Ttd
David Karel Halomoan
13520154