

FedTimeGAN: Geração de Dados Sintéticos de Séries Temporais via Aprendizado Federado de Redes Adversárias Generativas

Rômulo Moura¹, Matheus Monteiro¹, Ticianal C. da Silva², Atslands R. da Rocha¹

¹Departamento de Engenharia de Teleinformática
Universidade Federal do Ceará (UFC), Fortaleza, Brasil

²Instituto UFC Virtual, Universidade Federal do Ceará (UFC), Fortaleza, Brasil

{romulomoura, rochamatheus}@alu.ufc.br, ticianalc@insightlab.ufc.br, atslands@ufc.br

Abstract. *Monitoring using accelerometers helps evaluate the quality of human activity. As the data collected is typically time series, dependent on the size and consistency of the data, it may be necessary to use synthetic data generation techniques. Data privacy is essential for personal user data, even synthetic data, and sharing it to train models may not be appropriate. Federated learning allows you to train models without sharing data. This work proposes a model for generating synthetic time series data for human activity via federated learning and Generative Adversarial Networks (GANs), denominated FedTimeGAN. A comparative analysis was conducted between federated and centralized training for the model of the synthetic time series data generation. The results reveal that federated learning can be more effective in generating high-quality synthetic data for this application, with data from multiple users and maintaining privacy.*

Resumo. *O monitoramento por meio de acelerômetros é útil para avaliar a qualidade da atividade física das pessoas. Como os dados coletados são tipicamente séries temporais, dependentes do tamanho e da consistência dos dados, pode ser necessário utilizar técnicas de geração de dados sintéticos. A privacidade dos dados é essencial para dados pessoais de usuários, mesmo que sintéticos, e compartilhá-los para treinar modelos pode não ser adequado. O aprendizado federado permite o treinamento dos modelos sem compartilhar os dados entre os clientes. Neste trabalho, é proposto um modelo de geração de dados sintéticos de séries temporais para atividade física por meio de aprendizado federado e Redes Adversárias Generativas (GANs), denominado FedTimeGAN. Foi realizada uma análise comparativa entre treinamento federado e centralizado para o modelo de geração de dados sintéticos de séries temporais. Os resultados obtidos revelam que a abordagem federada pode ser mais eficaz na geração de dados sintéticos de alta qualidade para esse tipo de aplicação, com dados de múltiplos usuários, além de manter a privacidade.*

1. Introdução

A atividade física melhora os níveis de condicionamento físico e da qualidade de vida, além de poder auxiliar na redução da obesidade, evitando problemas clínicos. O monitoramento, através de dispositivos de Internet das Coisas (do inglês, *Internet of Things*,

IoT), tem se tornado importante para avaliar a qualidade da atividade física desenvolvida. O uso de acelerômetros para medir as atividades tem bastante utilidade e apresenta resultados promissores [Mo et al. 2012]. Os dados captados de acelerômetros são tipicamente tratados como séries temporais, são dependentes do tamanho e da consistência dos dados. Os recursos de hardware dos sensores e de rede são limitados, podendo trazer problemas em relação a quantidade e qualidade destes dados. [Ranja et al. 2023]. Desta forma, pode ser necessário utilizar técnicas de geração de dados sintéticos, contribuindo no volume de dados. Além disso, dados sintéticos contribuem para a privacidade dos usuários, principalmente quando relacionada a dados pessoais e sensíveis, como dados coletados por *smartwatches* ou outros dispositivos *IoT*.

A *Generative Adversarial Network* (GAN) [Goodfellow et al. 2014] é uma arquitetura de redes neurais em dois módulos, um gerador e um discriminador, que agem de forma adversária para gerar dados sintéticos de qualidade. A missão do gerador é gerar dados sintéticos que sejam indistinguíveis dos dados reais enquanto o discriminador recebe dados reais e sintéticos e classifica-os. As duas redes competem entre si e isso faz com que o gerador seja capaz de criar dados melhores. A TimeGAN [Yoon et al. 2019] é uma técnica de geração de dados sintéticos que utiliza redes adversárias para gerar dados de séries temporais realistas, por exemplo de acelerômetros. Ela age preservando as propriedades temporais e de dependências entre as variáveis dos dados originais. O diferencial da TimeGAN está na utilização da técnica *embedding learning*, que cria um espaço latente de menor dimensionalidade de forma que possam ser capturadas as características principais das séries temporais. O espaço latente é determinado a partir de uma rede neural recorrente (RNN). Dessa forma, a TimeGAN é capaz de gerar dados sintéticos de alta qualidade.

Uma das necessidades de dados pessoais de usuários de aplicações *IoT*, mesmo que sintéticos, é a privacidade dos dados. Assim, compartilhá-los, mesmo que com o intuito de treinamento, pode não ser adequado. Uma estratégia de treinamento que pode ser utilizada neste contexto é o aprendizado federado (do inglês, *Federated Learning*, FL) [McMahan et al. 2017]. O FL permite o treinamento colaborativo a partir de conjuntos de dados que estão distribuídos em diferentes clientes sem o compartilhamento destes dados e com a ajuda de um servidor de parâmetros. Um dos benefícios do FL é a preservação da privacidade dos dados e diminuição do consumo de recursos de rede, visto que os dados não são compartilhados ou enviados pela rede.

Neste contexto, o objetivo deste trabalho é investigar a capacidade da TimeGAN de gerar séries temporais de dados de monitoramento de atividades humanas, a partir de FL, além de sua capacidade de representar de maneira satisfatória o comportamento de múltiplos usuários. Com esse intuito, neste trabalho é proposto o FedTimeGAN, um modelo de geração de dados sintéticos de séries temporais via FL e GAN. Foram realizados experimentos com a TimeGAN para geração de séries temporais de dados de acelerômetros para múltiplos usuários, no quais a TimeGAN é treinada de forma centralizada (modo tradicional) e comparada com uma TimeGAN treinada com aprendizado federado (FedTimeGAN).

2. Trabalhos Relacionados

Em trabalhos anteriores, é possível observar estudos com o treinamento de GANs utilizando arquiteturas federadas. Em [Xin et al. 2020] é explicada uma das grandes dificuldades de se treinar GANs: o compartilhamento de dados. Em casos em que compartilhar os dados rompe com políticas de privacidade, é necessário encontrar uma maneira de treinar a rede sem que os dados sejam expostos a terceiros. Então, há a proposta de um modelo de treinamento federado para GANs focado em privacidade, o FL-GAN privado, um modelo de rede adversária geradora combinada com o limite de *Lipschitz* para gerar dados sintéticos de alta qualidade sem sacrificar a privacidade dos dados de treinamento.

Já no trabalho [Li et al. 2023] são discutidos os desafios no treinamento de GANs em ambientes federados, como a dificuldade de convergência do modelo agregado global, gerando dados sintéticos de baixa qualidade. Para isso, é proposto um modelo de treinamento federado para GANs, o IFL-GAN, que utiliza do indicador de performance MMD, a discrepância média máxima, em sua estratégia de agregação de pesos, mantendo cada GAN local com pesos diferentes, fazendo com que a GAN global venha a convergir de maneira mais rápida, mostrando uma evolução significativa no processo de treinamento de modelos generativos de maneira federada.

Em [Zhang et al. 2023] é demonstrada a integração do aprendizado federado com a técnica de *Few-Shot Image Generation* para lidar com a dificuldade de obtenção de dados para o treinamento de modelos de geração de imagem, dadas as mais modernas políticas de privacidade de dados. Através do treinamento distribuído desses modelos, é possível gerar imagens de alta qualidade. A solução proposta no artigo treina uma rede adversária generativa usando dados de alta qualidade de vários usuários utilizando o conceito de aprendizado federado, permitindo que vários usuários para fazer modelagem sem transferir dados para fora da área local, protegendo a privacidade. Um decodificador converte as amostras de imagens originais em vetores de recursos, adiciona ruído gaussiano aleatório e, em seguida, reconstrói as imagens originais.

O diferencial deste trabalho se dá pela utilização da TimeGAN com a intenção de produzir dados sintéticos de séries temporais com alta qualidade para atividades humanas com base em aprendizado federado, de modo que os dados pessoais não sejam compartilhados entre os clientes da rede. Uma análise comparativa é realizada entre o aprendizado tradicional (centralizado) em diferentes experimentos e o aprendizado federado.

3. FedTimeGAN: Aprendizado Federado na Geração de Dados Sintéticos de Séries Temporais

Neste trabalho é proposto o FedTimeGAN (Figura 1), no qual a arquitetura de FL é utilizada para treinar TimeGANs. O aprendizado federado possui dois artefatos centrais: (i) o servidor federado, responsável por realizar a agregação dos parâmetros dos modelos, e (ii) os clientes, no qual o treinamento local acontece. Os clientes são os sistemas finais da rede, como smartphones e desktops, dentre outros. O FL pode ser dividido nas fases:

1. O servidor envia os parâmetros do modelo inicial aos clientes e também as configurações de treino.
2. Os clientes iniciam o treinamento de acordo com os parâmetros e configuração de treino determinados pelo servidor. Ao fim do treinamento local, os clientes enviam os parâmetros do modelo treinado localmente para o servidor.

3. O servidor recebe todos os parâmetros enviados pelos clientes e realiza a operação de agregação. Existem vários métodos de agregação, o mais comum é o FedAvg [McMahan et al. 2017], que foi usado nesta proposta. A agregação transforma os vários parâmetros em apenas um, chamado de global.
4. O servidor envia o modelo global aos clientes para continuar o treinamento. Esta fase é repetida até que a quantidade de rodadas de treinamento definido na configuração de treino seja atingida.

A FedTimeGAN realiza o treinamento federado agregando no servidor o modelo gerador da TimeGAN, enquanto o discriminador permanece em treinamento local nos clientes. Desta forma, ao longo das rodadas de treinamento o modelo global do gerador da GAN melhora sua capacidade de criar dados sintéticos generalistas enquanto os discriminadores locais especializam-se em cada conjunto de dados.

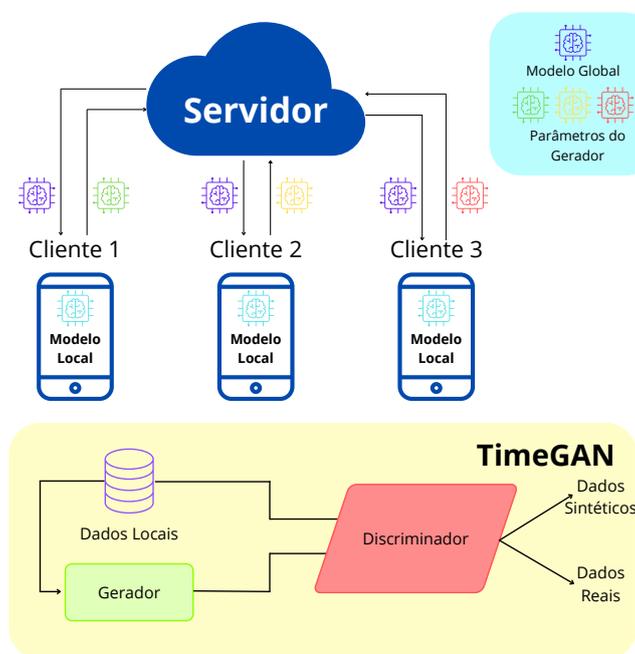


Figura 1. Arquitetura do FedTimeGAN

4. Experimentos e Resultados

Nesta seção, são descritos os diversos experimentos realizados para comparar o uso da rede TimeGAN com treinamento centralizado e federado. Em todos os experimentos realizados, os dados utilizados para realização deste trabalho são extraídos da base de dados *Physical Activity Monitoring* (PAMAP2) [Reiss 2012], que conta com 18 atividades físicas, realizados por 9 indivíduos, usando 3 unidades de medição inercial e um monitor de frequência cardíaca. Neste trabalho, foi utilizado um subconjunto da base de dados com 3 indivíduos (identificadores 101, 102 e 108), e com 3 atividades: *Standing* (em pé), *Lying* (deitado) e *Running* (correndo). Os dados são pré-processados com o propósito de tratar valores faltantes e *outliers*. Em nossa arquitetura, cada cliente possui dados de um único indivíduo (101, 102 ou 108) da base de dados PAMAP2.

A configuração dos hiperparâmetros utilizados na TimeGAN durante o treino (tanto centralizado como federado) envolve, 24 *timesteps*, *lambda* como 0.1, 64 dimensões escondidas, 3 camadas ocultas, ETA igual a 10, taxa de aprendizado de 0.001 e 64 para tamanho de lote. Os experimentos foram conduzidos em um computador com 32GB de RAM, com um processador AMD Ryzen 7 5800 8-core, onde foi criado um ambiente de experimentação, instanciando 3 clientes e um servidor de agregação. Os algoritmos foram implementados em utilizando a linguagem de programação Python (versão 3.10) e o Tensorflow 2.0 foi usado como *framework* de aprendizado de máquina.

Para a avaliação da qualidade dos dados sintéticos gerados pelos modelos TimeGAN e FedTimeGAN foi utilizado o *Maximum Mean Discrepancy (MMD)* [Tolstikhin et al. 2016]. Esta métrica calcula a discrepância entre duas distribuições através da diferença de suas médias amostrais. Quanto mais próximo de zero o valor de MMD melhor, indicando que as duas distribuições são próximas.

Para verificar a fidelidade dos dados gerados através dos modelos locais treinados de maneira federada, em relação aos dados gerados por cada paciente, foi aplicado o *Discriminative Score* [Yoon et al. 2019]. Esse índice mede a capacidade de um discriminador de identificar corretamente se sua entrada é real ou sintética. Para essa avaliação, o discriminador será implementado através de uma rede LSTM de duas camadas que será treinada para tentar diferenciar dados reais e dados sintéticos. Quanto mais próximo do valor 0, pior é a capacidade do discriminador de separar o que é real do que não é, indicando que os dados sintéticos são muito fiéis em relação os dados originais.

4.1. Geração de dados sintéticos com aprendizado centralizado

Três experimentos foram realizados para avaliação da TimeGAN com aprendizado centralizado conforme descrições a seguir. Em todos os experimentos foram usadas 1000 séries temporais como dados de treinos e 50 séries temporais como dados de teste. Além disso, foram realizadas 100 épocas de treinamento.

4.1.1. Experimento 1

Esse experimento visa investigar a geração de dados sintéticos criando-se um modelo para cada usuário. Para isso, vários modelos são treinados a partir de dados dos usuários, e os dados sintéticos são comparados com os dados originais (usando MMD) do conjunto de testes daquele mesmo usuário. As Tabelas 1, 2 e 3 exibem os valores de MMD para cada usuário nas atividades de *Standing*, *Lying* e *Running*, respectivamente. Os resultados de MMD na atividade de *Standing* mostram valores melhores para o usuário 102 do que para os outros usuários. Para a atividade de *Lying*, o usuário 101 obteve os melhores resultados, assim como para a atividade *Running*, o usuário 101 teve valores de MMD menores. Estes resultados demonstram que os conjuntos de dados usados neste trabalho não são uniformes, adicionando um desafio aos modelos para adaptar-se para uma generalização.

4.1.2. Experimento 2

Este experimento compara os dados sintéticos de um usuário A com os dados originais dos usuários B e C, respectivamente. O objetivo deste experimento é avaliar a capacidade

Tabela 1. Tabela de valores de MMD do Experimento 1 para atividade *Standing*

Usuário	mmd	mmd_x	mmd_y	mmd_z
101	0.191525	0.139392	0.003982	0.046769
102	0.002373	0.000002	0.000748	0.001639
108	0.018798	0.013298	0.002720	0.003034

Tabela 2. Tabela de valores de MMD do Experimento 1 para atividade *Lying*

Usuário	mmd	mmd_x	mmd_y	mmd_z
101	0.091491	0.001213	0.077205	0.017183
102	0.414419	0.332908	0.08388	0.031009
108	0.595658	0.499752	0.008267	0.144587

Tabela 3. Tabela de valores de MMD do Experimento 1 para atividade *Running*

Usuário	mmd	mmd_x	mmd_y	mmd_z
101	0.054237	0.039116	0.015176	0.001572
102	0.122027	0.005134	0.121847	0.000525
108	0.154664	0.164293	0.000182	0.030989

de um modelo treinado 100% com dados de apenas um usuário generalizar para outros usuários. Os resultados mostram (Tabelas 4 e 5) para as atividades *Standing* e *Lying*, que os modelos não são capazes de generalizar para outros usuários, não conseguindo obter valores de MMD parecidos com o Experimento 1. Já para a atividade *Running* (Tabela 6), nota-se que, apesar dos valores de MMD em relação ao Experimento 1 terem aumentado, os valores de MMD alcançados neste experimento indicam que nesta atividade um modelo treinado 100% com dados de um usuário é capaz de generalizar para outros usuários.

Tabela 4. Tabela de valores de MMD do Experimento 2 para atividade *Standing*

Treinado	Testado	mmd	mmd_x	mmd_y	mmd_z
101	102	0.142599	0.110460	0.032911	0.001897
101	108	0.427473	0.203409	0.054578	0.215542
102	101	0.146322	0.019177	0.040574	0.087628
102	108	0.418571	0.018377	0.144716	0.291672
108	101	0.182527	0.040636	0.053268	0.093200
108	102	0.534348	0.063181	0.209724	0.310183

Tabela 5. Tabela de valores de MMD do Experimento 2 para atividade *Lying*

Treinado	Testado	mmd	mmd_x	mmd_y	mmd_z
101	102	0.083254	0.016283	0.057988	0.011144
101	108	0.293302	0.009614	0.283554	0.007629
102	101	0.303255	0.194684	0.100297	0.038401
102	108	0.572488	0.301362	0.316699	0.024894
108	101	0.593768	0.398821	0.115829	0.190056
108	102	0.772361	0.571587	0.153926	0.178524

Tabela 6. Tabela de valores de MMD do Experimento 2 para atividade *Running*

Treinado	Testado	mmd	mmd_x	mmd_y	mmd_z
101	102	0.027716	0.024332	0.001523	0.002181
101	108	0.049718	0.034869	0.003040	0.004532
102	101	0.074350	0.010413	0.056964	0.010221
102	108	0.160623	0.010464	0.147547	0.007390
108	101	0.179043	0.159348	0.025295	0.017144
108	102	0.203715	0.190194	0.002423	0.044762

4.1.3. Experimento 3

Este experimento visa avaliar se o modelo é capaz de generalizar os dados sintéticos para todos os usuários se for treinado utilizando dados de todos eles. Neste experimento foi criada uma base de dados de treino formado por 33% de dados do usuário 101, 34% de dados do usuário 102 e 33% de dados do usuário 108. Neste experimento, nota-se que a atividade *Lying* (Tabela 8) obteve os menores valores de MMD. Os resultados são melhores em comparação aos Experimentos 1 e 2. Observa-se então que para a atividade *Standing* (Tabela 7), encontram-se resultados piores que o Experimento 1 e equivalentes ao Experimento 2. Para *Running* (Tabela 9), este experimento trouxe os piores resultados dentre todos os experimentos realizados.

Desta forma, pode-se observar que o Experimento 1 obteve os melhores resultados em duas das três atividades, indicando que um treinamento que cria um modelo para cada usuário pode ser a melhor solução para o treinamento centralizado de GANs de geração de dados sintéticos de atividade física.

Tabela 7. MMD do Experimento 3 para a atividade *Standing*

Usuário	mmd	mmd_x	mmd_y	mmd_z
101	0.312439	0.190703	0.195343	0.007041
102	0.419436	0.286298	0.063241	0.116278
108	0.506244	0.161486	0.368784	0.030306

Tabela 8. MMD do Experimento 3 para a atividade *Lying*

Usuário	mmd	mmd_x	mmd_y	mmd_z
101	0.010912	0.003985	0.003722	0.002163
102	0.012028	0.008169	0.000040	0.003832
108	0.104920	0.003794	0.096883	0.006394

Tabela 9. MMD do Experimento 3 para a atividade *Running*

Usuário	mmd	mmd_x	mmd_y	mmd_z
101	0.464693	0.004970	0.305389	0.243680
102	0.329726	0.005503	0.204855	0.174990
108	0.325894	0.007540	0.174336	0.184722

4.2. Geração de Dados Sintéticos com Aprendizado Federado - FedTimeGAN

Este experimento visa avaliar como o FL é capaz de treinar um único modelo global capaz de generalizar os dados sintéticos para todos os usuários. No experimento de FL foram utilizados treinamentos que totalizavam 100 épocas locais de treinamento para cada cliente, assim como no treinamento centralizado. A configuração escolhida foi de 5 rodadas federadas e 20 épocas locais. O algoritmo de agregação utilizado foi o *FedAvg*.

As Tabelas 10, 11 e 12 exibem os resultados de MMD para as atividades *Standing*, *Lying* e *Running* para cada usuário, respectivamente. Pode-se notar a priori que os valores de MMD obtidos neste experimento não são valores altos (apenas na atividade *Standing*, o MMD calculado ultrapassou 0.1). Nos resultados da atividade *Standing*, nota-se que os valores obtidos no treinamento federado são piores para os usuários 102 e 108 em comparação ao Experimento 1 do treinamento *centralizado*. Quando comparado aos outros experimentos do treinamento *centralizado*, o FL alcançou os melhores resultados. Nos experimentos da atividade *Lying*, os resultados de MMD alcançados pelo treinamento federado foram melhores que todos os experimentos do modo *centralizado*, com exceção para o usuário 102 do Experimento 3. Já para a atividade *Running*, o treinamento federado obteve resultados melhores que todos os experimentos realizados.

Desta forma, pode-se observar que o treinamento federado superou o treinamento centralizado, para a base de dados e rede neural selecionados neste trabalho, na maioria dos experimentos realizados, evidenciando a capacidade de generalização superior do treinamento federado.

Tabela 10. MMD Federado para atividade *Standing*

Usuário	mmd	mmd_x	mmd_y	mmd_z
101	0.028140	0.015457	0.002070	0.001115
102	0.120555	0.007637	0.050181	0.064913
108	0.107634	0.003075	0.037688	0.070454

Tabela 11. MMD Federado para atividade *Lying*

Usuário	mmd	mmd_x	mmd_y	mmd_z
101	0.017668	0.005778	0.006968	0.001993
102	0.020894	0.006162	0.014071	0.000056
108	0.039349	0.002572	0.036728	0.000185

Tabela 12. MMD Federado para atividade *Running*

Usuário	mmd	mmd_x	mmd_y	mmd_z
101	0.030222	0.008858	0.013836	0.002604
102	0.017062	0.005839	0.004536	0.001960
108	0.038585	0.010324	0.007901	0.006033

4.2.1. Avaliação Visual do Treinamento Federado - FedTimeGAN

A Figura 2 exibe as séries temporais reais e sintéticas para as três *features* dos dados (valores dos eixos x, y e z do acelerômetro), ao longo das 24 timesteps definidas no momento

de treino. Os dados reais plotados na Figura 2 são uma concatenação dos 3 usuários da atividade *Standing*. Todas as atividades geram dados sintéticos visualmente parecidos. Assim, é possível avaliar visualmente que a TimeGAN gera dados pouco variados em comparação com os dados reais, mas mantém o valor médio.

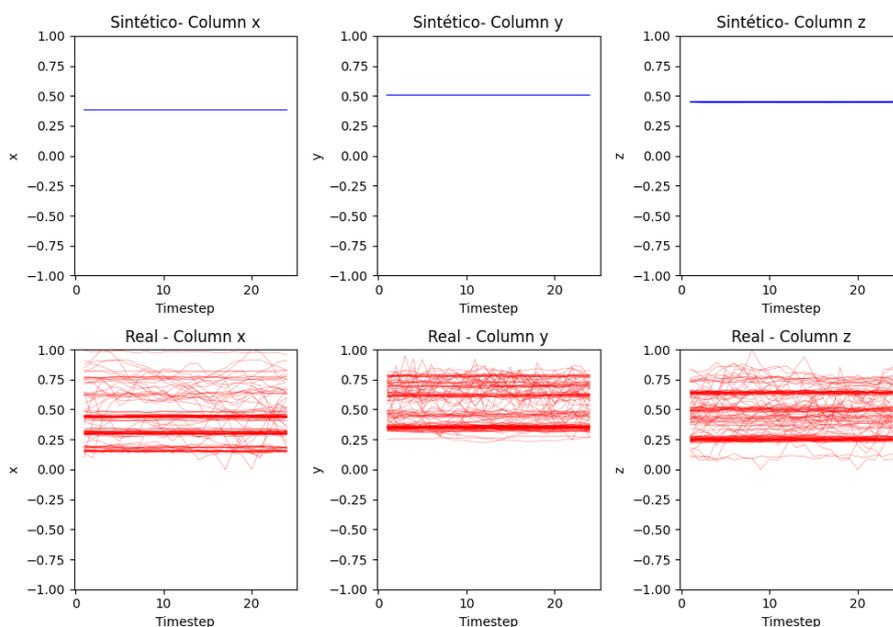


Figura 2. Séries temporais dos dados sintéticos e reais para atividade *Standing*

4.2.2. Avaliação através de *Discriminative Score*

A partir dos resultados da aplicação do FL (configuração de 20 épocas por *round*), foram calculados os *discriminative score* durante cada rodada de treinamento para a atividade *Standing*, obtendo os resultados mostrados na Tabela 13. A partir dos resultados, é possível observar que o discriminador, após o treinamento, não conseguiu identificar o que eram dados reais ou sintéticos. Pode-se entender que os geradores locais são capazes de gerar dados fiéis aos seus dados originais após participar de rodadas de FL.

Tabela 13. *Discriminative Score* para a atividade *Standing*

	<i>Discriminative Score</i>		
Rodada	Usuário 101	Usuário 102	Usuário 108
1	0.040000	0.020000	0.100000
2	0.000020	0.000400	0.000100
3	0.040000	0.080000	0.060000
4	0.020000	0.020000	0.020000
5	0.420000	0.000400	0.040000

5. Considerações Finais

Em contextos de monitoramento de pessoas por dispositivos IoT, que geralmente envolvem dados sensíveis, o uso de técnicas de geração de dados sintéticos pode contribuir

com a disponibilidade e volume de dados. Este trabalho propõe o FedTimeGAN, um modelo federado de treinamento de séries temporais utilizando a rede neural TimeGAN. Este trabalho combina técnicas de geração de dados sintéticos com arquitetura de aprendizado federado para gerar séries temporais mais precisas do que métodos centralizados de treinamento de modelos de aprendizado de máquina, e ao mesmo tempo, mantendo a privacidade. A proposta foi utilizada em um contexto de IoT com dados de atividade humana, gerando dados de acelerômetros. Os resultados experimentais obtidos revelam que o treinamento federado utilizando TimeGAN supera o treinamento centralizado de séries temporais com dados de múltiplos usuários.

Referências

- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.
- Li, W., Chen, J., Wang, Z., Shen, Z., Ma, C., and Cui, X. (2023). Ifl-gan: Improved federated learning generative adversarial network with maximum mean discrepancy model aggregation. *IEEE Transactions on Neural Networks and Learning Systems*, 34:10502–10515.
- McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR.
- Mo, L., Liu, S., Gao, R. X., John, D., Staudenmayer, J. W., and Freedson, P. S. (2012). Wireless design of a multisensor system for physical activity monitoring. *IEEE transactions on Biomedical Engineering*, 59(11):3230–3237.
- Ranja, F., Nababan, E. B., and Candra, A. (2023). Synthetic data generation using time-generative adversarial network (time-gan) to predict cash atm. In *2023 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, pages 418–423. IEEE.
- Reiss, A. (2012). PAMAP2 Physical Activity Monitoring. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C5NW2H>.
- Tolstikhin, I. O., Sriperumbudur, B. K., and Schölkopf, B. (2016). Minimax estimation of maximum mean discrepancy with radial kernels. *Advances in Neural Information Processing Systems*, 29.
- Xin, B., Yang, W., Geng, Y., Chen, S., Wang, S., and Huang, L. (2020). Private fl-gan: Differential privacy synthetic data generation based on federated learning. pages 2927–2931.
- Yoon, J., Jarrett, D., and Van der Schaar, M. (2019). Time-series generative adversarial networks. *Advances in neural information processing systems*, 32.
- Zhang, Y., Zhang, Q., Yang, M., Xiao, T., and Wang, Z. (2023). Fedgan: Federated gan for few-shot image generation. pages 1020–1024. Institute of Electrical and Electronics Engineers Inc.