Towards ‘Her’: An AI that Understands You

Eoin Lyness
Queen’s University Belfast
elyness02@qub.ac.uk

***Abstract*— As more of our personal data is collected by large tech companies, there has been an increasing need for an approach to software development and machine learning that focuses on the privacy of the user. We propose a system that using privacy-preserving techniques will monitor a user’s daily life and make suggestions in order to improve their life and their happiness. In addition to keeping sensitive information secure, this contrasts with how personal data is currently being used - primarily for advertising and delivering content - ultimately at the benefit of companies rather than the user. In this paper we consider recent approaches to privacy-preserving machine learning and provide an overview of the proposed system, first considering a traditional implementation then exploring an approach to transforming this initial system into one which is privacy-preserving. Our experiments compare these two approaches and demonstrate the feasibility and effectiveness of the system.**

***Index Terms*—artificial intelligence, privacy-preserving machine learning, recommendation systems.**

I. INTRODUCTION

In recent years privacy of personal data has become an increasing concern, most notably as dominant companies Amazon, Apple, Facebook and Google aim to collect increasingly more data about their users [1,2]. In the field of machine learning this has raised an important issue. Machine learning models thrive on large amounts of data, and to develop systems that study areas such as human behaviour or health requires sensitive data. Therefore, it is important to develop these systems with privacy of individuals in mind.

We propose a system which aims to influence and improve the everyday life of its users. By combining existing approaches in recommendation systems with privacy-preserving techniques, we outline a system that can learn how a user’s behaviour makes them feel by monitoring their daily life and with this information make suggestions that can affect their mood in a positive way. Where personal data is currently being used to benefit large companies, we instead aim to use it to benefit the user.

The system will collect data from the user’s device about the activities they perform as they go about their day. These activities will need to be labelled by the user so that they can be understood in a readable way and to ensure the accuracy of any ambiguous activities. The user will periodically tell the system how they are feeling, which will be used to understand how different activities affect their mood. With this information the system can learn what makes the user happy and make suggestions to try to improve their mood when necessary, through what it already knows about them and by using existing knowledge to predict what similar activities they may also enjoy. To create this system will require a scalable solution that is effective without the need for a large amount of data.

Our aim is that by creating the system with user privacy in mind, users can comfortably allow the system to capture and process data about them without fear that sensitive data will be exposed. We believe that this approach can help to lay the groundwork for future machine learning systems to be developed in this way.

First, we briefly introduce some of the techniques used in recent work to address privacy in machine learning and in the next section we discuss the literature implementing these techniques:

Homomorphic encryption [3] allows encrypted data to be sent to a server where computation can be performed without ever decrypting the data. The result produced is also encrypted.

Secure multi-party computation [4,5] is another encryption technique which may be seen as an improvement on the limitations of homomorphic encryption. Rather than sending all of the data to a server in order to perform the computation, secure multi-party computation allows the computation to be shared among the clients. Each client performs a share of the computation on its own encrypted data with the results of these individual calculations combined to achieve the overall result. The result may only be decrypted by combining the keys of each party.

Obfuscation [6,7] involves transforming the data in an attempt to prevent individual user data from being identified. This is often achieved through adding random noise to the data or masking sensitive information in such a way that the original data cannot be inferred, whilst preserving the overall statistics of the dataset.

Differential privacy is a similar technique to obfuscation in that it attempts to anonymise data through adding noise, allowing patterns in the data to be described without exposing information about individuals within the dataset. Differential privacy aims to ensure that no outputs become significantly more or less likely whether or not an individual is included in the dataset. Dwork et al. define ε-differential privacy according to Definition 1. [8,9]

Definition 1. A randomised function M with domain D is ε-differentially private if for all neighbouring datasets (differing on at most one element) $x, x' \in D$, and for all events (measurable sets) S in the space of outputs of M:

Eoin Lyness is with the School of Electronics, Electrical Engineering and Computer Science, Queen’s University Belfast, UK (e-mail: elyness02@qub.ac.uk).

$$Pr(M(x) \in S) \leq e^{\varepsilon} Pr(M(x') \in S) \qquad (1)$$

Federated learning [10] is a recent addition to privacy-preserving machine learning which aims to ensure privacy by locally training a model where the data is stored rather than sending data to a central server to perform the computation. The local updates to the model are aggregated by the central server to update the global model and thus individual data is not exposed.

II. Literature Review

An early approach to privacy-preserving machine learning was by Canny in [3], concerning recommendation systems using collaborative filtering. This approach used homomorphic encryption to perform the calculations and only decrypt the result, therefore preserving the privacy of the individual data. Whilst this is a promising approach, homomorphic encryption is a very computationally intensive task making it not well suited for scalable systems.

Polat and Du then proposed an obfuscation-based approach with collaborative filtering in [6], introducing randomness to the data to prevent the identification of individuals. J. Zhu et al. later implemented these randomisation techniques in [7], a privacy-preserving framework for QoS-based web service recommendation. Their experiments showed a common problem with many privacy-preserving approaches which is a trade-off between privacy and accuracy. Obfuscation presents limitations in particular with smaller datasets, on which it can significantly reduce the accuracy.

Dwork et al. proposed the concept of ε-differential privacy in [8], in which ε defines a privacy budget that determines how much noise is added to the data. As ε increases, the amount of privacy decreases. This work was implemented by Abadi et al. with deep learning [11] and by T. Zhu et al. with collaborative filtering [12], showing that increasing the privacy budget comes at a cost to the accuracy of the system. Like obfuscation, differential privacy can significantly reduce the accuracy with smaller datasets.

The term federated learning was first proposed by McMahan et al. in [10], which described a decentralised approach where the training occurs on the edge devices and the computed updates are aggregated by the global model. This approach ensures that sensitive data is secure by having it never leave the local device. In addition to the privacy guarantees, federated learning facilitates scalability by performing most of the computation on the edge devices and minimising communication with the central server. However, federated learning alone presents limitations in that information about the data may be inferred from a particular update. This issue was addressed by Bonawitz et al. [4] in introducing secure aggregation to the federated learning process. This technique uses secure multi-party computation to compute the sum of the updates from each device securely such that the individual updates are not revealed, only the overall result.

Mohassel and Zhang proposed SecureML [5], detailing the first efficient and scalable privacy-preserving machine learning protocols, using two-party computation with secret sharing, which improved existing privacy-preserving linear and logistic regression approaches by several orders of magnitude. In addition, this was the first privacy preserving system for training neural networks. As noted in the paper, earlier work in the field lacked implementation and thus SecureML was a significant milestone.

Developments in privacy-preserving machine learning ultimately led to the need for a practical and accessible framework to develop deployable systems using these techniques. Ryffel et al. proposed PySyft [13], a framework that extends the existing PyTorch library to implement federated learning, secure multi-party computation and differential privacy. Whilst their experiments showed a much slower training time and lower accuracy compared with a non-privacy-preserving approach it nonetheless was a significant milestone in presenting the first general framework for privacy-preserving machine learning, lowering the barrier to entry for further developments in the field.

With the rise of virtual assistants in our daily lives and increasing concerns of user privacy and the dominance of Big Tech, Campagna et al. proposed Almond [14], the first open-source, privacy-preserving virtual assistant. As discussed by Rafailidis and Manolopoulos [15], a gap exists between virtual assistants and recommendation systems. There have been many developments in the two but there is yet to be a true unified approach to the problem. Similarly, as discussed by Dwyer [1] and Smyth [2], the way personal information is currently being used by companies focuses on advertising and delivering content to the user, benefitting the companies rather than personally helping the user.

We propose a shift to a more personal focus for recommendation systems that seeks to improve the user's daily life. To do this poses new challenges not considered by prior work, in which the focus has very much been on hiding as much of the system and the data as possible to prevent personal information from being exposed. This system must be visible to the user in a way such that using their feedback it can improve over time and adapt to their needs, all while ensuring privacy of sensitive information is still preserved.

III. Problem Setting

A. *System Approach*

Fundamentally, the system will recommend activities to the user that it believes they will enjoy in order to make them happy. To achieve this the system will monitor the user's activity throughout each day and use this information to learn what the user's daily life looks like and how they feel about each of the activities they perform. In addition to this the system will use what it has learned about other users to recommend activities to the user that similar people to them enjoy. The data obtained about the user will need to be labelled describing what each activity is and how the user feels about it.

This problem can be thought of as being similar to that of movie recommendation systems, in which collaborative filtering is used to train the system based on a set of movies, and ratings of those movies by a number of users. The collaborative filtering system is used to recommend new movies to the user that they may enjoy based on how they have rated other movies and how other similar users have rated the movie. We can use this approach to the movie recommendation problem to fit our problem, where the activities performed by each user are movies, and the user's mood in relation to the activity is the rating. The activities must be discretised like a movie in order to be processed by the collaborative filtering system, which can be achieved by dividing the data gathered from the user into arbitrary, discrete time intervals, with each interval being treated as a movie. For our work we have chosen intervals of one hour, but future work should consider exploring alternative intervals or indeed alternative methods for discretisation.

If we treat each possible one-hour interval of data as being a separate movie, this would not be useful to the system as the same kinds of activities can be performed in many different ways and so it is highly unlikely that two intervals would contain identical data even if they contained the same activity. To deal with this we can make use of clustering on the hourly intervals so that similar intervals and consequently similar activities are contained within the same cluster, and thus each cluster can be considered as a movie in the movie recommendation analogy.

We have chosen to approach our work within the context of the movie recommendation problem due to the ease with which our problem can be translated to fit that of movie recommendations and indeed recommendation systems have been shown to be widely successful in commercial use, with services such as Netflix and YouTube. Similarly collaborative filtering naturally lends itself to an ever-growing set of both users and movies which is essential for our system as new users can join at any time and data will be constantly gathered from each of the users.

B. *Experimental Data*

Due to the restrictions of the General Data Protection Regulation (GDPR), it was not feasible for the experimental analysis of our work to make use of real user data. Consequently, the approach we have taken instead uses synthetic data which is generated to train and validate the system.

The generation of synthetic datasets has been explored by Walonoski et al. in creating synthetic health records [16] and by Yelmen et al. in creating artificial human genomes [17]. In these works, the synthetic data was generated using data contained in existing real datasets, ensuring that plausible data could be produced and allowing this to be evaluated against real data. It was not possible to take this same approach in our work and thus an alternative process was required. The synthetic data naturally needed to be generated from scratch with some degree of randomness to ensure variation in the dataset to aid the learning process.

Our approach makes use of a defined set of parameters indicating which activities would appear in the data as well as a set of user profiles containing mood ratings for each of the possible activities used to label the data produced. To produce usable data from these parameters, we created a set of templates that would be filled out with the activities and labels, with the templates structured in the format that would be output by the system with a real user. The labels used for identifying each of the activities were adapted from the activity identifiers used within the United Kingdom Time Use Survey 2014-2015 [18].

The proposed system is one which has not been created before and our goal with this work is to demonstrate a first example of how such a system would work. To achieve this, we focused on some of the core activities that many people might regularly be engaged with such as working, going for a walk, browsing the internet and social media, communicating with friends and family, listening to music, or watching videos and movies. To expand this further into a more complex and complete dataset would require a similar approach to that seen in the related work, wherein real data is obtained and used as a reference point for generating the synthetic data. However, the extensive ethics investigation that would be required to ensure compliance with GDPR is beyond the scope of this paper, thus we leave this as an area for future work.

IV. Overview of the Proposed System

In this section we consider an outline of the proposed system initially using non-privacy-preserving implementations of the key components. In the following sections we analyse the feasibility and effectiveness of this initial system and briefly explore and discuss how the system may be modified into one which preserves user privacy.

A. *Obtaining User Data*

The first component of the proposed system concerns the monitoring of a user's daily activity to gather as much information as possible about what a typical day in their life looks like. Both desktop and mobile tools were considered but we chose to focus our implementation in the form of a mobile application, as for most people a smartphone is their primary device and one which is generally always with them, thus providing a central source of information about their daily life.

We developed such an application for Android devices which makes use of many of the inbuilt features of the operating system to monitor the user's activity. The application runs in the background unobtrusively to the user, logging a history of their application usage, internet browsing, phone calls, texts and GPS location that is saved in JSON format on the device, with a separate file for each day.

Whilst we focus on only a basic implementation of this system here, future work should consider extending this functionality to involve deeper data logging such as through the use of APIs to retrieve user data from email, social media and streaming services. By gathering as much data as possible about the user's day, a more accurate and detailed picture about the life of the user can be developed, ensuring that meaningful recommendations can be made to them by the system.

B. *Processing the Data*

Before any learning can occur with the data, each log file produced is broken down into hourly blocks thus providing 24 log fragments for each day. The user will then be required to label the activities carried out during that part of the day as well as their mood during the time. To train the system the labelled hourly data is run through a clustering algorithm with the resulting clustered data used as the training data for the collaborative filtering model.

The distance function used within the clustering algorithm computes the similarity between two fragments by taking each activity within the one-hour interval in turn and comparing it to the corresponding activity in the other fragment. A percentage similarity of how many activities were the same within both

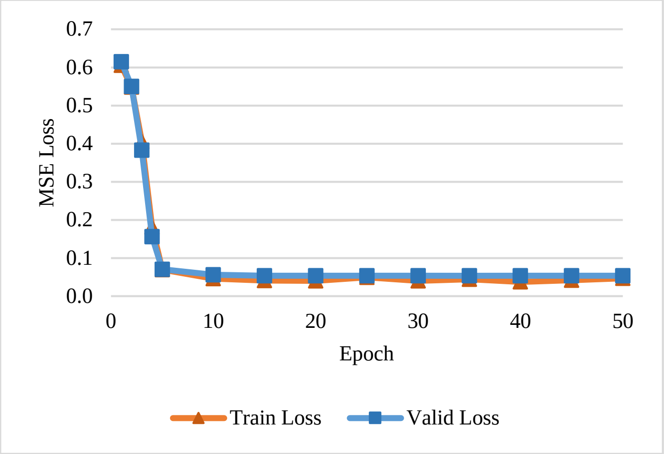

Fig. 1. Comparing training and validation MSE loss over 50 epochs with an 80/20 training/validation split on the dataset for the fastai collaborative filtering model (no privacy)

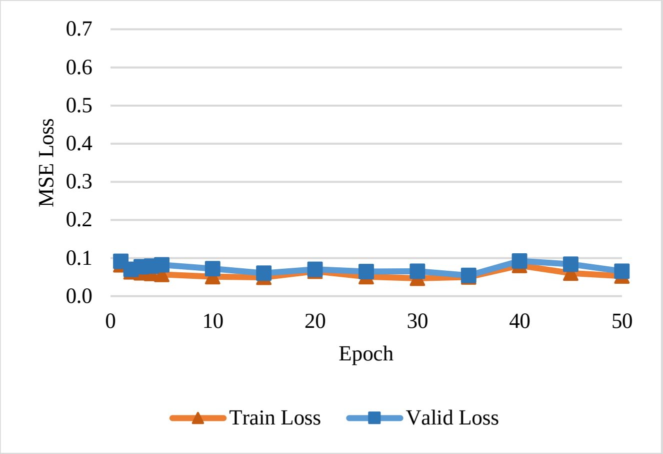

Fig. 2. Comparing training and validation MSE loss over 50 epochs with an 80/20 training/validation split on the dataset for the PySyft collaborative filtering model (privacy-preserving)

TABLE I
EXPERIMENTAL RESULTS

Model	Training MSE Loss	Validation MSE Loss	Avg. Training Time (s)
fastai	0.0469	0.0534	2.0
PySyft	0.0528	0.0652	26.2

Results for each model after training for 50 epochs.

hours is calculated, which is then converted into the resulting distance as a value between 0 and 1. The clustered data is then used by the collaborative filtering model to predict a user's mood rating for a particular cluster based on how similar users feel about that cluster.

V. RESULTS & DISCUSSION

To train the collaborative filtering model for our experiments, labelled synthetic data was generated and clustered for 1000 log files giving 24000 one-hour fragments. Two implementations of the model were explored; one with no privacy implemented using the fastai library, and the other privacy-preserving using federated learning implemented with the PySyft library. The PySyft model was trained with virtual workers corresponding to each user in the dataset performing updates to their own local copy of the model with these results aggregated into the global model after each iteration.

The same dataset was used to train both models with an 80/20 training/validation split. The training was run over 50 epochs measuring both training and validation mean squared error (MSE) loss on each iteration. Figure 1 illustrates the results over the 50 epochs using the fastai model, with Figure 2 showing the results for the PySyft model. Both models performed very well and produced similar results, with a training and validation MSE loss after 50 epochs of 0.0469 and 0.0534 respectively for the fastai model and 0.0528 and 0.0652 respectively for the PySyft model, as shown in Table I. Whilst the fastai model began with an initially significant loss that greatly decreased and levelled out after a number of iterations, the PySyft model started with a significantly smaller loss with minor fluctuations both lower and higher throughout the 50 epochs. This contrast in behaviour between the two models is likely a result of the difference in training processes, where the fastai model uses of an increasing learning rate with each epoch whilst the PySyft model uses a constant learning rate.

Whilst on average there was little difference in loss between the two models, the PySyft model performed much slower with an average training time of 26.2 seconds per epoch compared to 2 seconds per epoch for the fastai model, as shown in Table I. Additionally, this time does not consider any network latencies that may occur between local devices and the central server in a fully implemented federated learning system, thus slower training times would likely be seen in practice.

VI. TOWARDS A PRIVACY-PRESERVING SYSTEM

To transform the current system into one which is privacy-preserving, we believe it is vital that the implementation should adhere to the principles of Privacy by Design as proposed by Cavoukian [19], which outline the essential attributes that an effective privacy-based system should have. These are proactive privacy protection instead of remedial action after privacy violations have happened; privacy as the default setting; privacy embedded into the design; full functionality with full privacy protection; end-to-end security through the entire lifecycle; visibility and transparency; and respect for user privacy.

We propose making use of the PySyft framework to implement each of the system components and integrate them into a federated learning system which preserves user privacy. In addition to the PySyft implementation of the collaborative filtering model from the experiments, a PySyft implementation of the clustering algorithm would be developed, and these would be hosted on the central server using PyGrid – PySyft's server model – allowing federated learning to be carried out on user devices with these implementations. The KotlinSyft library can then be used to allow the models to be accessed and trained on Android devices, as well as the equivalent library, SwiftSyft, for iOS devices. With this implementation, the user's personal data will never leave their own device and consequently no third party outside of the user's device will be able to see their data.

The processing of the user’s log files would be carried out on their device and additionally the clustering and collaborative filtering would be carried out locally on each user’s device using a copy of the global model, with the individual updates being sent to the server to update the overall model and ensure consistency for the results of every user. The significantly slower training speed with PySyft compared to a non-privacy-preserving implementation is largely irrelevant as a training update can occur only once per day and this can be carried out overnight when the user’s devices are not in active use.

This implementation of the system would preserve user privacy outside of their device, but there is a still a question concerning privacy within their device. Transparency of the system to the user is essential as the user needs to have access to their own data in order to label it, meaning that the data needs to be exposed in some way on the device and arguably this violates privacy as a third party may be able to intercept this data from the device. A solution to this could be to encrypt the local data, only decrypting it when it is necessary to expose it to the user, namely for labelling purposes. This could indeed be taken a step further by adding an authentication layer to the application to ensure that it is indeed the user attempting to access the software.

We believe that the use of PySyft components presents a promising architecture for implementing a privacy-preserving system and indeed that the use of federated learning is the best suited approach to the problem. However, an area for future work would be to explore and compare alternative implementations with other existing privacy-preserving frameworks such as TensorFlow Federated.

VII. Evaluation & Future Work

The system presents several areas for future work. It would be beneficial to obtain real data for use in evaluating the system as we have been unable to prove from our experiments how a real-world implementation of this system would perform under the nuance of real data. Indeed, future work should explore how effectively the system is able to adapt to an ever-expanding universe of both activities and users.

Another area for future work is in dealing with the complexity of mood and indeed the many complex situations that may arise in the lives of users while using the system. External factors can affect the enjoyment someone has of a particular activity temporarily which is not always immediately clear. The system should be able to learn and adapt to complex situations such as this and indeed it would benefit the user for the system to be able to recognise potential causes for a change in mood and use this information to adapt its response to the user.

In addition to expanding both the amount and the complexity of information that is gathered about the user’s day, future work should look at extending the functionality of the system to make it more useful and effective for the user. One approach to this would be to alleviate the need for the user to label their data with each activity they have been performing by enabling the system to learn what the user is doing at any particular time. This may be implemented through the use of segmentation on the log data produced, so that the system can determine when different activities are taking place and classify them accordingly. Removing the need to manually label the data could help to enhance the privacy-preserving nature of the system as the data would no longer need to be exposed to the user. However, it would still be beneficial to maintain a level of transparency with the user and receive feedback on the predictions made so that the accuracy and quality of the system can improve over time.

VIII. Conclusion

We have proposed a system for monitoring the daily activities of a user to learn about their lives and how they feel about the different tasks they perform. We have demonstrated the feasibility and effectiveness of this system with our experiments and discussed an approach to implementing the system in a privacy-preserving architecture, protecting the user’s sensitive data. By providing a first solution to this problem and highlighting several potential areas for future work, we lay the groundwork for further developments in this domain. Our aim is to use machine learning as a means of improving people’s lives by using their personal data for their own benefit rather than as a product for advertisers and large companies. With this aim in mind, we believe that a greater trust between human and AI can be developed, enabling people to view it as a friend, and to lessen the caution and hostility that many have about sharing their personal data.

References

[1] C. Dwyer, "Privacy in the Age of Google and Facebook," in *IEEE Technology and Society Magazine*, vol. 30, no. 3, pp. 58-63, Fall 2011.

[2] S. M. Smyth, “The Facebook Conundrum: Is it Time to Usher in a New Era of Regulation for Big Tech?,” *International Journal of Cyber Criminology*, vol. 13, no. 2, Art. no. 2, 2019.

[3] J. Canny, "Collaborative filtering with privacy," *Proceedings 2002 IEEE Symposium on Security and Privacy*, Berkeley, CA, USA, 2002, pp. 45-57.

[4] K. Bonawitz et al., “Practical secure aggregation for privacy-preserving machine learning,” in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191.

[5] P. Mohassel and Y. Zhang, "SecureML: A System for Scalable Privacy-Preserving Machine Learning," *2017 IEEE Symposium on Security and Privacy (SP)*, San Jose, CA, 2017, pp. 19-38.

[6] H. Polat and Wenliang Du, "Privacy-preserving collaborative filtering using randomized perturbation techniques," *Third IEEE International Conference on Data Mining*, Melbourne, FL, USA, 2003, pp. 625-628.

[7] J. Zhu, P. He, Z. Zheng and M. R. Lyu, "A Privacy-Preserving QoS Prediction Framework for Web Service Recommendation," *2015 IEEE International Conference on Web Services*, New York, NY, 2015, pp. 241-248.

[8] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *TCC'06: Proceedings of the Third conference on Theory of Cryptography*. Berlin, Heidelberg: SpringerVerlag, 2006, pp. 265-284.

[9] C. Dwork, “Differential privacy”, in *ICALP’06*. Springer-Verlag, 2006, pp. 1-12.

[10] B. McMahan, E. Moore, D. Ramage, S. Hampson and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, 2017, pp. 1273-1282.

[11] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar and L. Zhang, "Deep Learning with Differential Privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, New York, NY, USA, 2016, pp. 308-318.

[12] T. Zhu, G. Li, Y. Ren, W. Zhou and P. Xiong, "Differential privacy for neighborhood-based Collaborative Filtering," *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013),* Niagara Falls, ON, 2013, pp. 752-759.

[13] T. Ryffel et al., "A generic framework for privacy preserving deep learning," arXiv *preprint arXiv:1811.04017*, 2018.

[14] G. Campagna, R. Ramesh, S. Xu, M. Fischer, and M. S. Lam, "Almond: The Architecture of an Open, Crowdsourced, Privacy-Preserving, Programmable Virtual Assistant," in *Proceedings of the 26th International Conference on World Wide Web*, Perth, Australia, 2017, pp. 341–350.

[15] D. Rafailidis and Y. Manolopoulos, "The Technological Gap Between Virtual Assistants and Recommendation Systems," *arXiv*, pp. arXiv–1901, 2018.

[16] J. Walonoski et al., "Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record", *Journal of the American Medical Informatics Association* vol. 25, no. 3, 2018, pp. 230–238.

[17] B. Yelmen et al., "Creating artificial human genomes using generative neural networks," *PLoS genetics*, vol. 17, no. 2, Art. no. 2, 2021.

[18] J. Gershuny and O. Sullivan, "United Kingdom Time Use Survey, 2014-2015," UK Data Service, 2017, [Online] Available: http://doi.org/10.5255/UKDA-SN-8128-1

[19] A. Cavoukian, "Privacy by Design: The 7 Foundational Principles," Information and Privacy Commissioner of Ontario, Canada, Toronto, ON, 2009.