DOI 10.31891/2307-5732-2024-331-36
УДК 004.056, 004.8

ONAI MYKOLA
National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute"
https://orcid.org/0000-0002-4938-8355
e-mail: onay@pzks.fpm.kpi.ua
SEVERIN ANDRII
National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute"
https://orcid.org/0009-0009-1366-8054
e-mail: severinandrey97@gmail.com

ARCHITECTURE OF A SOFTWARE SYSTEM FOR SOLVING THE CLASSIFICATION PROBLEM BASED ON PRIVATE DATA

Data analysis and artificial intelligence systems are becoming widely used in various spheres of human life. This is confirmed by more typical cases of their use, in particular, the selection of recommendations for the user in e-commerce, the detection of spam in e-mail services, and the moderation of user comments; as well as cases of personal use of such tools (for example, chatbots ChatGPT, Google Bard, Microsoft Copilot have appeared and gained significant popularity in the last two years). One of the key elements of such systems is data, which is necessary for training and testing software systems of intelligent data analysis. A significant amount of diverse data contributes to the construction of a software system with high accuracy. Considering this, the task of choosing and preparation of datasets that can be used in the construction of such systems is important. One of the difficulties in this task is the presence of private information in the datasets, which limits their use for systems of intelligent data analysis.

The paper is devoted to the development of the software system architecture for solving the classification problem based on private data. The existing methods and architectural approaches for privacy-preserving in machine learning were considered. The architecture of the software system was proposed, the characteristic feature of which is the protection of private datasets by using of functional encryption, which allows to increase the number of datasets for training publicly available data analysis and artificial intelligence systems. The proposed architecture of the software system is based on the client-server architecture and functional encryption. The components are a classifier, a generator of encryption keys, and modules of functional encryption and decryption. Prospects for further research were discussed.

Keywords: privacy-preserving machine learning, software architecture, functional encryption, classification problem.

ОНАЙ МИКОЛА, СЕВЕРІН АНДРІЙ
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

АРХІТЕКТУРА ПРОГРАМНОЇ СИСТЕМИ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ КЛАСИФІКАЦІЇ НА ОСНОВІ ПРИВАТНИХ ДАНИХ

Системи аналізу даних та штучного інтелекту набувають значного поширення у різних сферах людського життя. Це підтверджують, як більш типові випадки їх використання, зокрема підбір рекомендацій для користувача у електронній торгівлі, виявлення спаму в сервісах електронної пошти та модерація коментарів користувачів; так й випадки особистого використання таких інструментів (наприклад, впродовж останніх двох років з'явились й набули значної популярності чатботи ChatGPT, Google Bard, Microsoft Copilot). Одним з ключових елементів таких систем є дані, які є необхідними для навчання та тестування систем інтелектуального аналізу даних. Значна кількість різнопланових даних сприяє побудові програмної системи з високою точністю.

Стаття присвячена розробленню архітектури програмної системи для вирішення задачі класифікації на основі приватних даних. Розглянуто існуючі методи для збереження приватності в машинному навчанні. Запропоновано архітектуру програмної системи характерною особливістю якої є захист приватних наборів даних, шляхом функціонального шифрування, що дозволяє збільшити кількість наборів даних для навчання загальнодоступних систем аналізу даних і штучного інтелекту.

Ключові слова: машинне навчання із збереженням приватності, архітектура програмного забезпечення, функціональне шифрування, задача класифікації.

Problem statement

The main data source for software solutions in the artificial intelligence field is the real world. Also, there are software methods of data generation, the purpose of which is to reproduce certain features of the data. However, even though the amount of data is growing rapidly, it often contains at least part of private information, which limits its use for software systems of intelligent data analysis. The protection of private data is important because the loss of privacy can lead to negative consequences (in particular, to various crimes).

One of the most common tasks of data analysis and artificial intelligence systems is the classification task. It is being solved in many areas, including the financial transactions classification (for fraud detection or market analysis), the medical data classification (for disease diagnosis), and the user requests classification in technical support systems. Classification consists of determining whether input data belongs to one of several predefined classes (categories) based on the characteristics of such data.

Therefore, the task of improving private datasets processing in systems that use artificial intelligence to solve the classification problem is urgent.

The purpose of this work is to design the architecture of a software system for solving the classification

problem based on private data.

The object of research is the processes of private data processing in software systems of intelligent data analysis.

The subject of the research is models, methods, and approaches to the development of software systems for protecting private datasets in classification tasks.

Literature Review

The main ways to protect private datasets in machine learning tasks are the following: the generation of synthetic datasets, processing of private datasets (data anonymization, differential privacy, homomorphic encryption) and federated learning [1-11]. Most of these methods involve centralized processing of system training information. This means that the data must be gathered and stored on one device (it can be on-premises or cloud server) that is used to train the system of intelligent data analysis. However, there is one decentralized architectural approach – federated learning. It was proposed by the Google researchers [7, 9]. The idea of this approach is to train an artificial intelligence algorithm on many end devices or servers that contain local datasets that remain on the device during training. In this case, local datasets are not shared between devices. This approach differs from traditional centralized machine learning methods, where all data samples are uploaded to a single server, as well as from more classic decentralized approaches, which assume that local data samples are distributed evenly across devices.

Consider an example of such an architectural approach to learning, which is shown in Fig. 1 [9]. Let's it is necessary to solve the problem of classification of various geometric shapes (circle, square, rhombus, oval, etc.). When using federated learning, the current version of the model is stored on a server, for example in the cloud. The user device downloads an initial version of the model to their device (such as a phone or tablet) and refines it by learning a local dataset (block A in Fig. 1), then summarizes the changes in the model weights and sends them to the cloud as a small update. Once this update, sent in encrypted form, arrives at the server, it is immediately averaged with updates from other users (block B in Fig. 1), and the weight of the overall model is improved (block C in Fig. 1). Then this procedure is repeated. However, all data used for training remains on the user's device, and updates are not stored in the cloud.

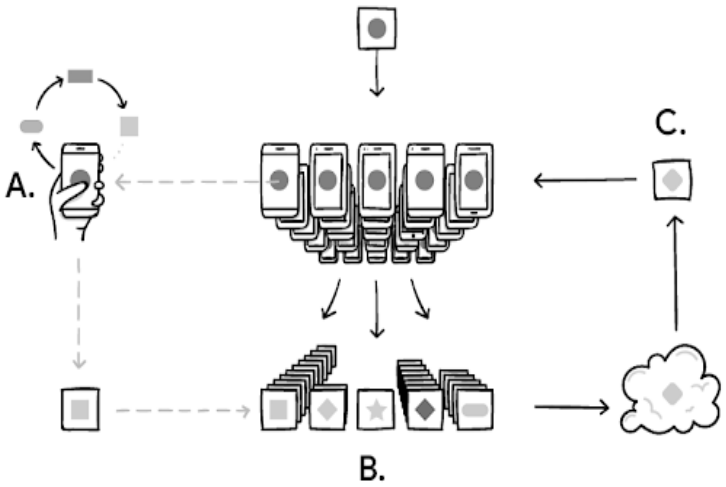

Fig. 1. An architectural approach to federated learning (Federated Learning)

It is worth noting that with this approach, local models can show both better and worse prediction results compared to the joint model, since the training on local devices can be uneven (with different amounts of data), but due to the averaging of updates, the joint model is constantly improving, as well as local models that showed worse than average results. With the help of federated training, it is possible to create models while guaranteeing privacy, while the latency and use of server resources will be lower, since a large part of the training will be done on local devices. This approach allows not only to update the joint model, but also to improve the local model with minimal delay. However, the approach of federated learning does not solve all the problems of machine learning systems, because with its help it is not possible, for example, to train a model to recognize different breeds of dogs by training the model on labeled examples. In addition, a significant part of the data is centralized and stored in cloud storage, so in such cases this approach is not appropriate. Therefore, the federated learning method is a reliable and accurate method, it does not distribute the local training data, however, it can be used if there are at least a few independent users who have enough training data. Giving this, the development of an alternative decentralized architecture that minimizes the considered limitations is an urgent task.

Presenting Main Material

The main task of the software system for solving the problem of classification based on private data is to preserve the privacy of system users. Consider this, the architecture of the software system was developed, which is shown in Fig. 2. The proposed architecture of the software system is based on the client-server architecture and functional encryption.

Client-server architecture is a model of interaction of computer systems, where computing resources and functions are divided between client and server components. The basic idea is that clients (users) interact with servers (usually over a network) to gain access to resources provided by the server. This may include retrieving data, performing calculations, storing information, etc. An important feature of the client-server architecture is the separation of duties between client and server programs, which simplifies the development, maintenance, and scaling of systems. In addition, this architecture allows support for various types of clients (e.g. computers, mobile devices) and servers, making it widely used in modern software [12].

Functional encryption is a type of encryption that provides more precise control over access to encrypted data compared to traditional encryption methods [13, 14]. The purpose of functional encryption is to selectively disclose certain information or functions contained in encrypted data to authorized parties while preserving the confidentiality of the rest of the data. When traditional data encryption is used, the owner of the key can decrypt the entire content of the message. However, functional encryption allows the owner of the key to compute only certain functions on the encrypted data without revealing the entire original content. This is achieved by associating different keys with different functions.

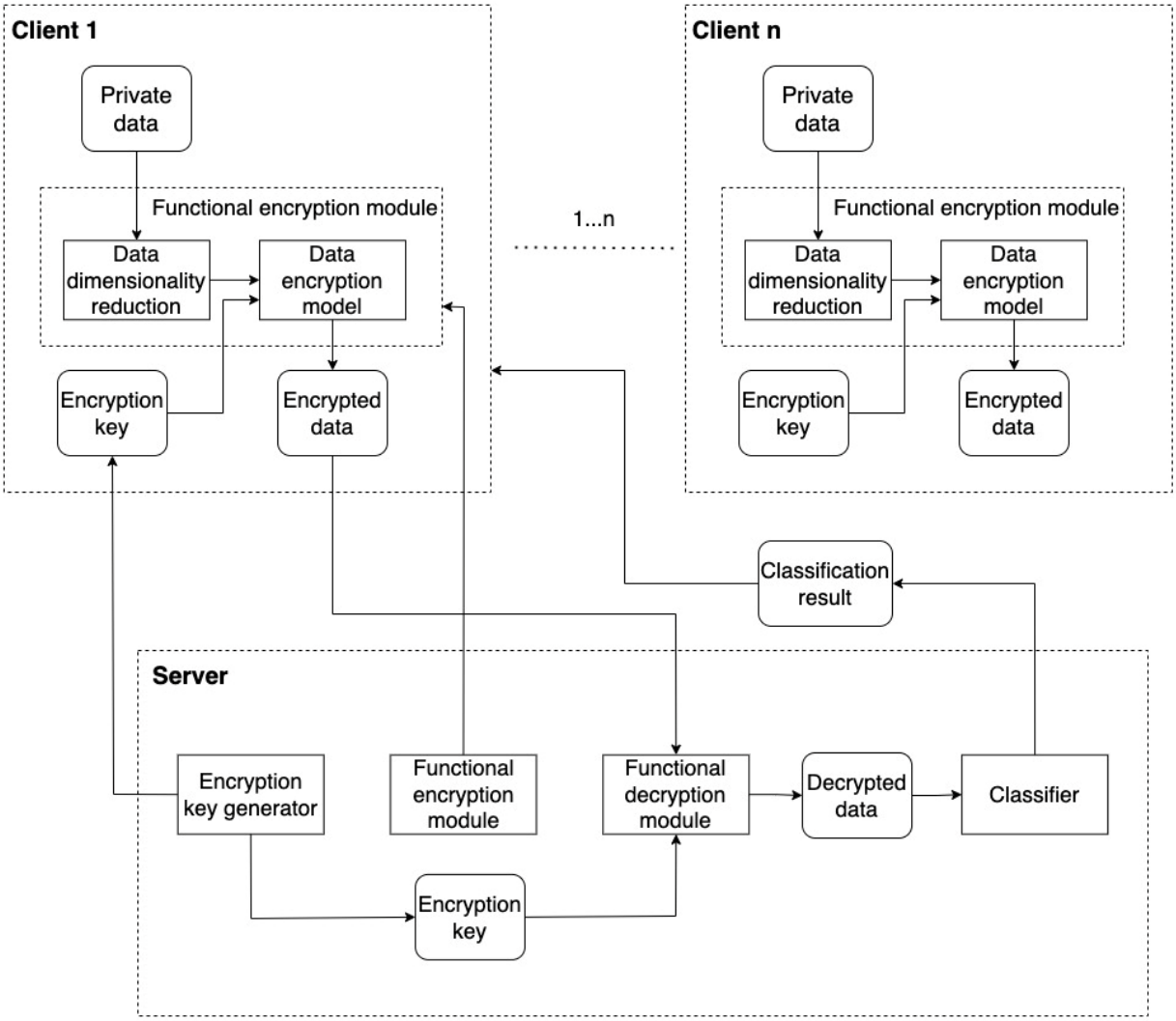

Fig. 2. Architecture of a software system for solving the classification problem based on private data

In the developed architecture, it is proposed to host the encryption and the classifier models on a server (that can be located locally or in a cloud environment) since the training of such components requires a significant number of resources. Also, in this case, the server allows many users to use such models. According to Fig. 2, the server contains an encryption key generator, a functional encryption module, a functional decryption module, and a classifier. The server processes encrypted and decrypted data and generates encryption keys.

At the first stage, the client downloads a functional data encryption module, which is pre-trained on data of a given dimension. The functional encryption module implements the functional encryption method consists of a data dimensionality reduction submodule and a data encryption model. The user's private data remains with the client. If necessary, it can de-anonymized beforehand. After that, the dimension of the private data of the loaded encryption module is reduced, where the dimension of the input data is reduced, and its encryption is carried out using the software model (for example, mentioned in papers [15, 16]). The encryption key used in the encryption model is generated on the server and sent to the client. The result of this step is encrypted data.

After that, the server receives the encrypted data from the client and the encryption key from the encryption key generator. This data is processed by the functional decryption module, resulting in decrypted data that can be used to train, test, and use the classifier. The result of the classifier is sent to the client.

The number of clients, according to the developed software system architecture, is unlimited. Also, there can be one client. This differs from federated learning methods where it is important to have at least a few independent users who have enough training data. The client structure is the same, and the encryption key is generated separately for each client. According to the developed architecture, private data is not distributed to the server, due to which the privacy of system users is preserved.

Conclusions

The architecture of the software system for solving the classification problem based on private data is proposed, the characteristic feature of which is the protection of private datasets using functional encryption that occurs on the client side and allows to increase the number of datasets for training publicly available data analysis and artificial intelligence systems.

Further research can be focused on the development of different implementations of the modules that are used in this architecture (in particular, the generator of encryption keys, the module of functional encryption and decryption), as well as experimental studies of the application efficiency of the proposed architecture for different datasets and data classifiers.

References

1. Xu R., Baracaldo N., Joshi J. Privacy-preserving machine learning: Methods, challenges and directions. arXiv preprint arXiv:2108.04417. 2021. DOI: 10.48550/arXiv.2108.04417.

2. Lauter K. Faculty Summit 2017: Private AI. Microsoft Research. 2017. https://www.microsoft.com/en-us/research/wp-content/uploads/2017/07/Private_AI_Kristin_Lauter.pdf.

3. Emam K. E., Mosquera L., Hoptroff R. Practical Synthetic Data Generation. 2020. 163 p. O'Reilly Media, Inc. ISBN 978-1492072744.

4. Rubinstein I. S., Hartzog W. Anonymization and risk. Washington Law Review. 2016. № 91. C. 704–760.

5. Dwork C., Roth A. The Algorithmic Foundations of Differential Privacy. Foundations and Trends® in Theoretical Computer Science. 2014. Vol. 9, № 3-4. P. 211–407. DOI 10.1561/0400000042.

6. Minelli M. Fully homomorphic encryption for machine learning. 2018. 157 p.

7. Brendan McMahan H., Moore E., Ramage D. Communication-efficient learning of deep networks from decentralized data. 2016.

8. Konečný J., McMahan B., Ramage D. Federated Optimization: Distributed Optimization Beyond the Datacenter 2015. https://arxiv.org/pdf/1511.03575.pdf.

9. Brendan McMahan H., Ramage D. Federated Learning: Collaborative Machine Learning without Centralized Training Data. H. Brendan McMahan. 2017. https://ai.googleblog.com/2017/04/federated-learning-collaborative.html.

10. Severin A.I., Onai M.V. Metody zberezhennia pryvatnosti v mashynnomu navchanni. Herald of Khmelnytskyi National University. 2023. № 6. S. 274-280/ DOI: 10.31891/2307-5732-2023-329-6-274-280.

11. Onai M.V., Severin A.I. Kompleksnyi porivnialnyi analiz metodiv zberezhennia pryvatnosti v mashynnomu navchanni. Aktualni zadachi suchasnykh tekhnolohii: zb. tez dopovidei KhII mizhnar. nauk.-prakt. konf. Molodykh uchenykh ta studentiv, (Ternopil, 6-7 hrudnia 2023), Tern. natsion. tekhn. un-t im. I. Puliuia [ta in.]. Ternopil: FOP Palianytsia V. A., 2023. S. 406-407.

12. Berson A. Client/server Architecture/ 1996. 569 p. McGraw-Hill, Inc., Professional Book Group 11 West 19th Street New York, NY, United States/ ISBN: 978-0-07-005664-0.

13. Panzade P., Takabi D. SoK: Privacy Preserving Machine Learning using Functional Encryption: Opportunities and Challenges. 2022. https://arxiv.org/pdf/2204.05136.pdf.

14. Boneh D., Sahai A., Waters B. Functional Encryption: Definitions and Challenges. Lecture Notes in Computer Science. Springer: Berlin, Heidelberg. 2011. Vol. 6597. p. 253-273. DOI 10.1007/978-3-642-19571-6_16.

15. Abadi M., Andersen D. G. Learning to Protect Communications with Adversarial Neural Cryptography. 2016. https://arxiv.org/abs/1610.06918.

16. Coutinho M., Albuquerque R. D., Borges F. Learning Perfectly Secure Cryptography to Protect Communications with Adversarial Neural Cryptography. Sensors (Basel). 2018.