Towards efficient and secure analysis of large datasets

Stelvio Cimato
Dipartimento di Informatica
Università degli studi di Milano
Milan, Italy
stelvio.cimato@unimi.it

Stefano Nicolò
Dipartimento di Informatica
Università degli studi di Milano
Milan, Italy
stefano.nicolo@studenti.unimi.it

***Abstract*—One of the promises of the "big data" revolution is that trough the analysis of large datasets people will benefit from the solution to many different problems obtained by the deployment of advanced machine learning models. One of the challenges of this standard approach, is that information needs to be centralized on the data center or the machine where the training phase is performed, posing many concerns about privacy.**

In this paper we take a step towards secure and efficient processing of distributed large datasets, where original data reside at different locations and are processed in a privacy preserving way. In particular we rely on the available technologies to achieve the secure design of a machine learning model by performing the training phase on encrypted data. The case study we examine is focused on the forecasting of energy production by wind farms situated in different locations. We show in detail how the machine learning model is created on the basis of the available datasets, we compare the results with the ones produced by the previous models, and discuss also their performances.

***Index Terms*—machine learning; privacy preserving techniques; secure multi-party computation**

I. Introduction

The large amount of data that are being produced and stored at an unprecedented rate is more and more used to develop machine learning models that can produce some improvements in different aspects of our live, from medical screening and disease outbreak [14], [15], to malware detection [16] and astronomical observations [21].

Usually, machine learning models are developed following a centralized approach, where data are accumulated in large datasets that are provided to a central server to create and train the machine learning model. This approach, however, poses a number of challenges, especially when large-scale collections include sensitive data [17]. Indeed, the storage of large amount of data at a central server may lead to a violation of users' privacy and augment the risks of releasing personal data, infringing also current regulations such as GDPR [10].

An alternative approach, recently proposed and supported by Google researcherswho named it Federated Learning (FL) [13], [22], consists of a machine learning setting where multiple clients collaborate to the training phase under the orchestration of a central provider, while the data to be analyzed is kept at the production sites. The local machine learning models can then be combined contributing to the update of the global federated model, that in turn can provide feedbacks to the local machines. In this way, many of the challenges of the centralized approach are addressed, such as privacy, locality and data ownership protection.

The way data and models are combined leverages on secure multi-party computation techniques (MPC) [7] and further advances the research on privacy preserving data analysis, started more than 30 years ago and relying on different cryptographic primitives, such as secret sharing, homomorphic encryption or predicate encryption [4], successfully deployed in fields such as data outsourcing [11], [18], private set intersection [3], and many others.

In this paper we describe the application of current available privacy preserving machine learning techniques to the forecasting of energy production in renewable energy scenario. The case study we consider is focused on the analysis of data coming from three sites where wind farms are located. We describe in detail how the machine learning models are created and trained on encrypted data and compare the resulting forecasts with the ones developed in traditional setting, giving also the analysis of their performances in terms of computation time.

The paper is organized as follows: in the next section we describe the energy market scenario, while in Sec. III we describe the libraries deployed for the creation of the machine learning model. In Sec. IV we describe the datasets used for the training of the resulting model. We discuss the forecasts resulting from the execution of the model on the considered datasets in Sec. V as well as the comparison with the forecasts resulting from the execution of the traditional models. Finally, performances are analyzed in section VI while related works and conclusions are reported in section VII and VIII, respectively.

II. The energy market

Forecast mechanisms play an important role in the energy market where they serve as a prerequisite for the maximization of the stability and the reliability of the energy grid, placing an accurate balance between production demand and electricity supply. For the producers, penalties may be imposed if the production is different from what previously estimated in order to maintain an equilibrium among the different suppliers. In case of an overestimated energy production, the operator may

need to buy quotes from other suppliers in a more expensive market. So in the event of errors in production estimates, suppliers could be forced to buy in an unfavorable period of the market.

The increase in the percentage deriving from renewable sources - mainly photo-voltaic and wind - which has occurred in recent years, and also the objectives imposed at both national and international level require a further increase in renewable sources in the coming years and consequently there will be the need for more reliable estimates. Optimizing energy production from these sources adds challenging elements, their production being subordinated to external factors that are difficult to control such as the presence and strength of the wind, temperature, cloudiness, etc., causing large fluctuations in actual production that are generally difficult to predict.

In the electricity market there are two trading paradigms: the long-term one, more stable, and the short-term one (daily or hourly horizon) that is exposed to high price variability. For the long-term market, trades are made using so-called forward contracts. Due to the characteristic dependence of the renewable energy sector on atmospheric variables, therefore the lower reliability of the forecast and the difficulties in planning tends to neglect the forward contracts market in favor of the short-term one. In this context, three ways of buying and selling are possible:

1) *First Day Market (MGP)*: counter-parties exchange hourly blocks of energy production for the next day; the market opens every trading day at 8:00 and closes at 11:55. The time interval between closing and reopening of the market is justified by the need for technical times for the delivery of the goods.
2) *Intraday market (MI)*: used in the event of emergencies; operators can use this market to make the necessary adjustments to their planning. To this end, the intraday market, organized into seven blocks, opens when the market for the previous day is closed.
3) *Market for Dispatching Service (MSD)*: in this market, the operator of the electricity grid deals with the procurement of the resources necessary for the management and control of the system. Especially in the renewable energy market this is a fundamental tool for managing differences in estimated production.

III. Preliminaries

In this section we discuss the software libraries that have been chosen to develop the machine learning model. Currently there are a number of libraries enabling the design of machine learning algorithms, we selected *Tensorflow* and *Crypten* on the basis of their characteristics and performance.

A. Tensorflow

TensorFlow [5] is a library capable of exploiting the *GPU* and enabling the creation of neural networks with multiple levels and neurons. TensorFlow has been developed and used by Google company, and was released for free as open-source in 2015, with the purpose of creating a standard for the exchange of ideas, speeding up research on machine learning algorithms, and providing a tool to allow the reuse of already designed neural networks or ML models. To make complex calculations, this library uses flow graphs, where information passing through nodes if processed to design a neural network. TensorFlow therefore, taking advantage of these features and the possibility of processing the data through a single API on various distributed CPUs (server, desktop, Mobile, etc.) allows researchers to make their products evolve faster and faster, and share the code more rapidly [5].

B. Pytorch and Crypten

PyTorch has been implemented by Facebook developers and is based on the same operations implemented in Tensorflow (flow graph logic, Tensors, etc.). For its simplicity and efficiency, it has gained popularity among developers and has been implemented in a number of ML projects. Pytorch provides also an excellent management of neural networks and the possibility of exploiting GPU computation power.

CrypTen is a Privacy Preserving Machine Learning framework written using PyTorch that allows researchers and developers to train models using encrypted data [9]. The framework allows the implementation of secure multi-party computation in a transparent way and offers excellent tools for integrating it into numerous projects. Crypten provides an additional library, *MPCTensor* element, that is nothing more than a tensor where the values are encrypted using a secret sharing protocol for multi-party computation.

IV. The machine learning model

A. The datasets

The datasets include a number of parameters that have been measured for each of the three wind farms under analysis. The datasets have an hour as granularity and have been collected for a period of about two years, from 2017 to the first months of 2019. The considered features are:

- *Asset_availability:* maximum power that can be produced by the wind farm as a whole during the referenced hour: it varies from the nominal maximum (61800, 37500 and 30000 following for *Site1*, *Site2* and *Site3*, respectively) up to a lower value non-negative, determined by the number of turbine stops (for example for maintenance reasons).
- *WTG_active:* number of active turbines, helping to integrate the reasons for the reduced availability of resources.
- *Grid_limitation:* limit threshold provided by the electricity grid operator in KW. The maximum value equals the maximum of the nominal power of the whole site.
- *Avg_power:* average power generated per turbine in KW.
- *Power:* power generated in the referenced hour by the entire wind farm: it is the target variable, and the unit of measurement has been converted in KW.

The data provided by the customer have been further analyzed and filtered to eliminate also some discrepancies or inconsistencies, and then processed to apply some simple machine learning algorithm. Such data have also been integrated with

the weather data collected at each site (provided by a regional agency), where every morning the forecast is produced for the remaining hours of the current day plus the following 24.
The new dataset, used to integrate the original one, has the variables listed below:

- *Wind_speed:* wind speed in meters per second.
- *Wind_u, Wind_v:* components of the wind vector, representing the direction and speed of the movements.
- *Wind_direction:* direction of origin of the wind, from zero to 360 degrees.
- *Temp:* temperature in degrees centigrade.
- *Pressure:* atmospheric pressure in pascal.
- *Cloud:* cloud coverage, represented as a percentage.

B. Neural network model

Based on the previous parameters, we developed different neural networks and analyzed their results. The one reporting satisfactory results and performance can be classified under these 4 different configurations:

1) **NN-64-32:** neural network with 2 hidden layers, with 64 and 32 nodes respectively.
2) **NN-128-64:** neural network with 2 hidden layers, with 128 and 64 nodes respectively.
3) **NN-256-128:** neural network with 2 hidden layers, with 256 and 128 nodes respectively.
4) **NN-128-64-32:** neural network with 3 hidden layers, with 256, 128 and 32 nodes respectively.

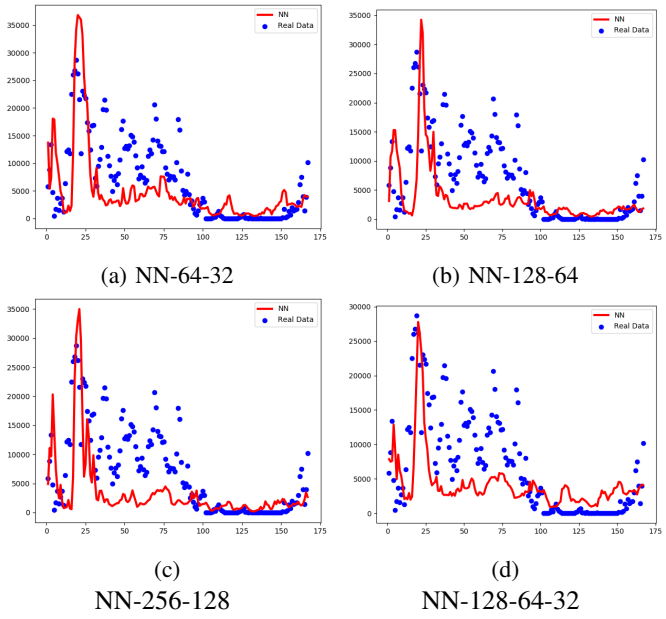
Obviously each neural network has in addition to the listed hidden layers an input layer, with a node for each feature, and an output layer with a single node. Figure 1 reports the behaviour of the considered networks for the dataset under analysis.

It can be immediately noticed that the results produced by the various configurations are very similar. All four networks, for example, predict the 2 peaks present at the beginning of the dataset quite well, the *NN-128-64-32* NN with greater precision without overestimating the highest peak. All curves then settle down, approaching the forecast towards zero in the second half of the dataset, in this case the *NN-128-64* network does it better. As far as the first network in the figure is concerned, the *NN-128-64*, rises more by approaching the real data in the 50-70 segment. To have more objective data than visual inspection, we consider the Mean Squared Error (MSE) to select the best performing network. In the following table we report the execution time and the MSE obtained by each NN., showing the values observed during the various executions:

	NN-64-32	NN-128-64	NN-256-128	NN-128-64-32
No. of Iterations	115	175	97	200
MSE error	0.0475	0.0473	0.0459	0.0523
Training Time (s)	48.06	121.44	253.56	442.26
Forecast Time (ms)	25.46	21.92	83.83	111.29

TABLE I: Training and forecast times for the different NNs

From these data it can be observed that the number of iterations varies greatly according to the configurations. An important point is obtained by observing the value of MSE; for almost all NNs this value is very similar except for the network with 3 different hidden layers, which is higher. The most important comparison is played on performance, reminding that these values have been measured on encrypted values and on a history of only 3 months for a target of 7 days.



(a) NN-64-32 (b) NN-128-64

(c) NN-256-128 (d) NN-128-64-32

Fig. 1: Forecasts obtained from the different neural networks tracing the energy production (y-axis) in the time period (x-axis).

V. Experimental results

In this section we examine the results obtained by applying the proposed machine learning models to the forecasting of the energy production in the considered wind farms. In the three sites, different models have been adopted in the past, and a number of data have been collected. These data are then compared with the results obtained by the application of the neural networks described in the previous section, working directly on plain-text data, and on encrypted data as well.
We quickly report which models have been considered optimal by the machine learning experts for each wind farm in the previous times:

- **Site1:** for this site only the linear regression model *Support Vector Regression* has been implemented.
- **Site2:** for this site the best results have been obtained by applying an arithmetic mean computed on the results of the following models: *Ordinary Least Squares Regression*, *Lasso regression* and *Support Vector Regression* . In addition, the ensemble learning method, *Random Forest* has been also used.
- **Site3:** for this site, the best result has been obtained by applying an arithmetic mean on the following models: *Lasso regression*, *Support Vector Regression* and *Random Forest*.

All the results shown in the following are therefore based on these approaches and the neural network with 2 hidden layers and 64 and 32 nodes (selected on the basis of results and performances). For all sites, the forecasts made over two different periods are reported:

- **Period 1:** From 22 to 24 October 2018
- **Period 2:** From 2nd to 4th November 2018

A. Forecasts for Site1

The wind farm located at Site1 has a maximum nominal power of 30000 KWh of producible energy.

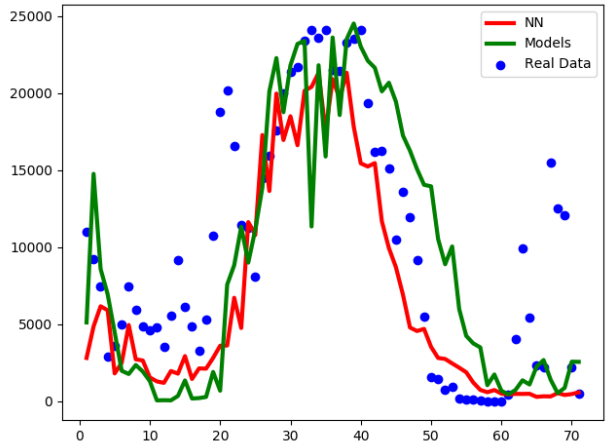


Fig. 2: Results obtained in the first period from ML and NN encrypted for Site1

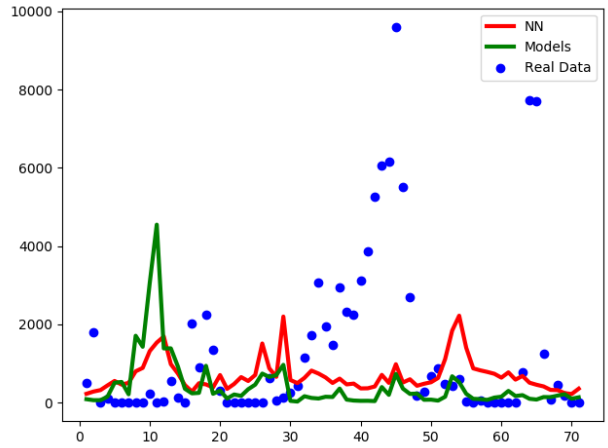


Fig. 3: Results obtained in the second period from ML and NN encrypted for Site1

From both the graphs and the computation of the Mean Absolute Error (MAE), it appears that the generated forecast curves are very similar, but the encrypted neural network obtains slightly better results in almost all the days taken into consideration. Considering the two periods under observation, in the first one the neural network obtains a score of well *0.7* less than the traditional model, while in the second period both models obtain good similar results.

	MAE Encrypted NN	MAE ML Models
MAE forecast 10/22	4,847	5,233
MAE forecast 10/23	3,770	3,592
MAE forecast 10/24	3,372	5,404
MAE forecast 11/02	0.779	0.959
MAE forecast 11/03	2,316	2,472
MAE forecast 11/04	1,182	0,909
MAE first period	4.005	4.734
MAE according to period	1,429	1,454

TABLE II: Mean Absolute Error reported for the NN and the ML models for Site1

B. Forecast for Site2

The wind farm located at Site2 has a maximum nominal value of 37500 kWh. Differently from the previous case, the data science team found it more efficient to use an approach with different models, having a more flexible method that exploits the different strengths of the various models.

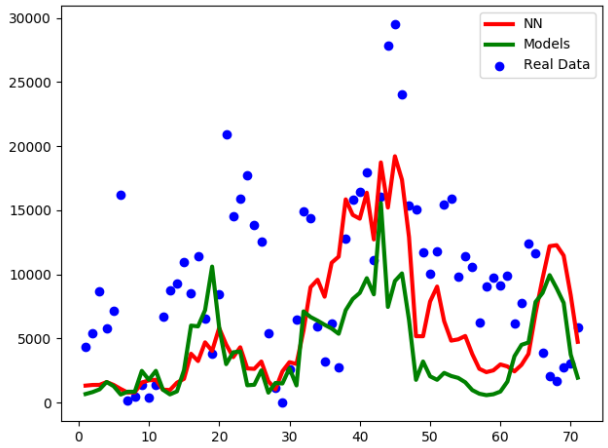


Fig. 4: Results obtained in the first period from ML and NN encrypted for Site2

	MAE Encrypted NN	MAE ML Models
MAE forecast 10/22	6,012	6,323
MAE forecast 10/23	5,071	6,888
MAE forecast 10/24	6,252	7,270
MAE forecast 11/02	12,159	11,304
MAE forecast 11/03	8,177	6,965
MAE forecast 11/04	7,546	6,936
MAE first period	5,772	6,821
MAE according to period	9,319	8,423

TABLE III: Mean Absolute Error reported for the NN and the ML models for Site2

The results obtained for this wind farm confirm once again that the results have a very similar curve and are therefore comparable. Compared to the previous case, here we get a quite different result between the two periods taken into consideration. In the first period, the encrypted neural network achieves better results on all the days taken, reaching a gap

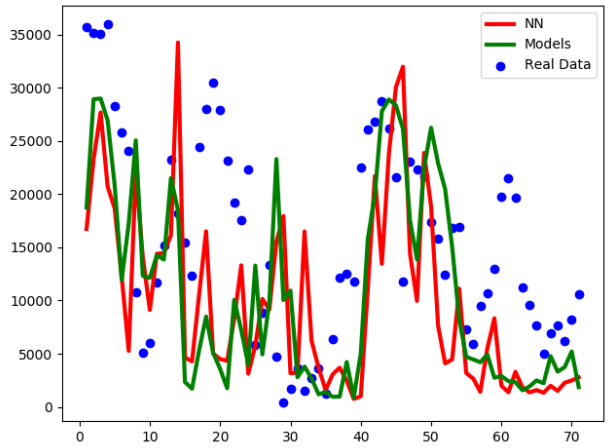


Fig. 5: Results obtained in the second period from ML and NN encrypted for Site2

of almost 2 points on the second day of the first period. In the second period, however, it is the machine learning models that obtain the most efficient results on all the scheduled days. In fact, looking at the overall result of the periods, there are inverse results; this indicates how both approaches guarantee a fairly efficient coverage of the problem but depending on the trend of the dataset in a given observation period an approach turns out to be better, and in other data windows vice-versa.

C. Forecasts for Site3

This wind farm located at Site3 has a nominal power of 61800 KWh. Also in this case, the data science team has selected an approach with the use of multiple models.

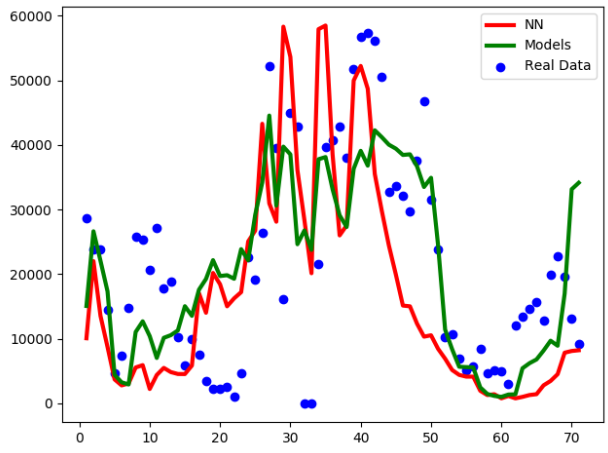


Fig. 6: Results obtained in the first period from ML and NN encrypted for Site3

Once again, the fact that both approaches follow very similar forecasting trends is confirmed. In the first period it is possible to observe that there are peaks in the center of the graph that are better covered by the neural network operating on

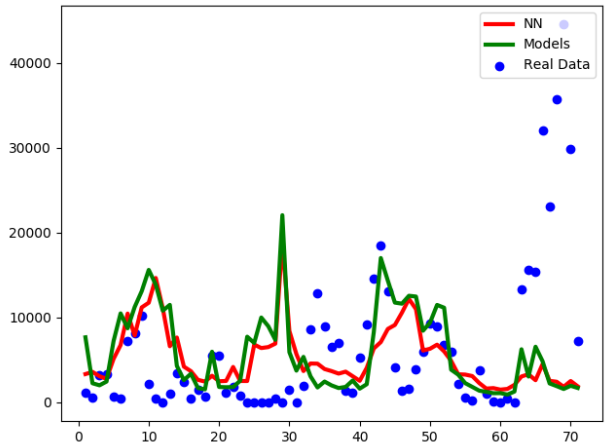


Fig. 7: Results obtained in the second period from ML and NN encrypted for Site3

	MAE Encrypted NN	MAE ML Models
MAE forecast 10/22	11,197	10,138
MAE forecast 10/23	15,953	12,031
MAE forecast 10/24	9,580	6,692
MAE forecast 11/02	3,631	4,244
MAE forecast 11/03	6,314	6,617
MAE forecast 11/04	9,325	9,106
MAE first period	12,281	9,662
MAE according to period	6,383	6,621

TABLE IV: Mean Absolute Error reported for the NN and the ML models for Site3

encrypted data. The calculation of the MAE shows that better results are obtained by the machine learning models in the first period, while in the second period, the results of the encrypted neural network are slightly closer to the real points.

By observing all the results produced on the various wind farms, it can be observed that the neural networks operating on encrypted data represent an excellent alternative to the traditional machine learning models, obtain valid forecasts wingith the advantage of having high security for the privacy of customer data.

VI. Performance Analysis

The developed neural networks show good performance in terms of both computational resources and computation time. The two tables below summarize the time needed comparing three different approaches to perform the model training and obtain the target forecasts. The first table shows the time recorded for the training phase on a dataset including a 3 months period, carried out by the machine learning models, the neural network on plain-text data, and the neural network on encrypted data, respectively (all reported data are expressed in seconds).

The data in the table above is stable for all three wind farms, apart from the time needed for Site1, that is reduced since it relies on one simple ML model (*SVR*), as explained in section V). The neural network operating on encrypted data takes approximately 6 times the time required for ML models

	Models ML	NN Plaintext	NN Ciphertext
Site1	0.335	12.410	35.745
Site2	5,149	12,647	33,214
Site3	5.873	12.703	35.239

TABLE V: Computation time in seconds for the training phase

and 3 times the time required for the neural network operating directly on plain-text data.

	Models ML	NN Plaintext	NN Ciphertext
Site1	4.278	25.954	156.996
Site2	10.209	26.560	98.223
Site3	10,605	25,389	127,111

TABLE VI: Computation time in seconds for the forecasts

The table above, shows the time observed with the same parameters but on a training history of a whole year. The observation of performances using all the available history shows that times are growing much more for the neural network trained on encrypted data, as expected since the execution of MPC protocol has an overhead. In this case, the neural network takes $10 - 12$ times more to carry out training and forecasting in the case of complex ML models and about 40 times more in the case of Site1 that adopts a simple model. In any case, times are still acceptable, requiring less than 3 minutes to produce a valid forecast, fully respecting the privacy of the datasets used in input.

VII. Related works

Cryptographic methods for computing in the encrypted domain have been proposed in the last forty years, trying to address benchmarking and classification techniques respecting data privacy [1], [19], [20]. Many proposals have been based on results coming from advancements in multi-party secure computation [7] and on homomorphic encryption [8]. Privacy preserving machine learning models have been proposed to solve classification problems deploying neural network where the training and computation phases are securely executed [2], [6]. Secure federated machine learning frameworks have been proposed by Google researchers starting from 2016 as possible solutions to address many of the challenges related to the security and privacy of the data [12], [13], [22].

VIII. Conclusions

Despite many researches performed in the last 40 years, only recently the advancements on both theoretical and technological aspects for privacy preserving machine learning are allowing to obtain practical results. In this paper we have described how the deployment of current frameworks for the design and the execution of neural networks can be efficiently deployed to solve the forecasting problem in the energy market scenario. The next step is to further develop the proposed framework and deploy the developed models following the secure federated learning approach.

Acknowledgment

The authors acknowledge the support of the EU Horizon 2020 research programs Concordia (Project-ID No. 830927).

References

[1] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD 00, page 439450, New York, NY, USA, 2000. Association for Computing Machinery.

[2] M. Barni, C. Orlandi, and A. Piva. A privacy-preserving protocol for neural-network-based computation. In *Proceedings of the 8th workshop on Multimedia and security*, pages 146–151, 2006.

[3] C. Blundo, E. D. Cristofaro, and P. Gasti. Espresso: Efficient privacy-preserving evaluation of sample set similarity. *J. Comput. Secur.*, 22(3):355–381, 2014.

[4] C. Blundo, V. Iovino, and G. Persiano. Predicate encryption with partial public keys. In S. Heng, R. N. Wright, and B. Goi, editors, *Cryptology and Network Security - 9th International Conference, CANS 2010, Kuala Lumpur, Malaysia, December 12-14, 2010. Proceedings*, volume 6467 of *Lecture Notes in Computer Science*, pages 298–313. Springer, 2010.

[5] G. Brain. Tensorflow. 2016.

[6] H. Chabanne, A. de Wargny, J. Milgram, C. Morel, and E. Prouff. Privacy-preserving classification on deep neural network. *IACR Cryptology ePrint Archive*, 2017:35, 2017.

[7] I. Damgård, V. Pastro, N. Smart, and S. Zakarias. Multiparty computation from somewhat homomorphic encryption. In *Annual Cryptology Conference*, pages 643–662. Springer, 2012.

[8] J. Fan and F. Vercauteren. Somewhat practical fully homomorphic encryption. *IACR Cryptology ePrint Archive*, 2012:144, 2012.

[9] D. Gunning, A. Hannun, M. Ibrahim, B. Knott, L. van der Maaten, V. Reis, S. Sengupta, S. Venkataraman, and X. Zhou. Crypten: A new research tool for secure machine learning with pytorch, 2019.

[10] A. Gupta. How federated learning is going to revolutionize AI. https://towardsdatascience.com/how-federated-learning-is-going-to-revolutionize-ai-6e0ab580420f, 2019.

[11] M. A. Hadavi, R. Jalili, E. Damiani, and S. Cimato. Security and searchability in secret sharing-based data outsourcing. *International Journal of Information Security*, 14(6):513–529, 2015.

[12] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[13] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[14] V. Lampos, A. C. Miller, S. Crossan, and C. Stefansen. Advances in nowcasting influenza-like illness rates using search query logs. *Scientific reports*, 5:12760, 2015.

[15] J. Paparrizos, R. W. White, and E. Horvitz. Screening for pancreatic adenocarcinoma using signals from web search logs: Feasibility study and results. *Journal of Oncology Practice*, 12(8):737–744, 2016.

[16] N. Peiravian and X. Zhu. Machine learning for android malware detection using permission and api calls. In *IEEE 25th international conference on tools with artificial intelligence*, pages 300–305. IEEE, 2013.

[17] J. Rodriguez. The challenges of centralized AI. https://towardsdatascience.com/the-challenges-of-decentralized-ai, 2019.

[18] M. Sepehri, S. Cimato, E. Damiani, and C. Y. Yeun. Data sharing on the cloud: A scalable proxy-based protocol for privacy-preserving queries. In *2015 IEEE TrustCom/BigDataSE/ISPA, Helsinki, Finland, August 20-22, 2015, Volume 1*, pages 1357–1362. IEEE, 2015.

[19] J. Vaidya, C. Clifton, M. Kantarcioglu, and A. S. Patterson. Privacy-preserving decision trees over vertically partitioned data. *ACM Trans. Knowl. Discov. Data*, 2(3):14:1–14:27, 2008.

[20] J. Vaidya, H. Yu, and X. Jiang. Privacy-preserving SVM classification. *Knowl. Inf. Syst.*, 14(2):161–178, 2008.

[21] J. van der Gucht, J. Davelaar, L. Hendriks, O. Porth, H. Olivares, Y. Mizuno, C. M. Fromm, and H. Falcke. Deep horizon: A machine learning network that recovers accreting black hole parameters. *Astronomy & Astrophysics*, 636:A94, Apr 2020.

[22] Q. Yang, Y. Liu, T. Chen, and Y. Tong. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.*, 10(2), Jan. 2019.