Protection des données d'entraînement pour l'apprentissage statistique

Eloïse Berthier

Direction générale de l'Armement
eloise.berthier@m4x.org

Résumé Les modèles d'apprentissage statistique sont susceptibles d'exposer les données qui ont été utilisées lors de leur entraînement. Ce phénomène doit être pris en compte pour qualifier le niveau de sensibilité d'un modèle. La notion de confidentialité différentielle, créée à l'origine pour la protection de la vie privée, répond partiellement à cette problématique. En particulier, il est possible d'adapter le processus d'apprentissage de façon à vérifier certaines propriétés de confidentialité. Lorsque les données sensibles sont distribuées sur plusieurs machines, des processus cryptographiques permettent d'entraîner conjointement un modèle sans en partager les données d'entraînement.

Keywords: confidentialité · apprentissage · protection · secret.

Les algorithmes d'apprentissage statistique ont permis des avancées significatives dans le traitement des données massives. Leur efficacité est directement liée à la qualité des données utilisées lors de l'entraînement et à leur adéquation avec les données qui seront utilisées ensuite en production. Dans le secteur de la défense, ces algorithmes devront nécessairement s'appuyer sur des données potentiellement classifiées pour atteindre une efficacité maximale.

1 Nécessité d'une protection des données d'entraînement

Une première question est celle du **niveau de sensibilité qui doit être accordé à un modèle entraîné à partir de données classifiées**. Plus précisément, si un modèle a été entraîné avec des données provenant de plusieurs sources avec différents niveaux de protection, à quel point expose-t-il ses données d'entraînement ? Est-il possible de modifier le processus d'apprentissage pour forcer le modèle produit à respecter un certain niveau de confidentialité ?

Plusieurs études empiriques apportent une réponse alarmante à la première question. En effet, on observe un phénomène de mémorisation involontaire de certains exemples d'entraînement par le modèle entraîné. Cet effet est d'autant plus marqué que le modèle dispose d'une grande capacité, comme c'est le cas notamment des réseaux de neurones. Un exemple frappant est celui d'un réseau de neurones entraîné sur du texte anglais qui a mémorisé des mots aléatoires, comme des mots de passe, insérés dans le jeu d'entraînement [3]. Deux grands

types d'attaques de rétro-ingénierie à partir d'un modèle ont été étudiées : l'attaque d'appartenance, qui consiste à deviner efficacement si une donnée particulière a été utilisée ou non lors de l'entraînement, et l'attaque de reconstruction, qui consiste à reconstruire une donnée d'entraînement ou certains de ses attributs. De telles attaques peuvent être menées soit en disposant des paramètres complets du modèle (*white box attack*), soit en disposant seulement d'un accès au modèle par des requêtes d'inférence (*black box attack*). Une approche possible pour répondre à ces enjeux est celle de la confidentialité différentielle appliquée à l'apprentissage statistique. C'est l'objet de la section partie de cet article.

Un second enjeu est celui du **partage de données entre plusieurs acteurs pour entraîner conjointement un modèle**. Cette question a son importance, y compris dans le domaine civil, pour lequel la Direction Générale des Entreprises a lancé en 2018 un appel à manifestation d'intérêt pour soutenir des projets de mutualisation de données entre acteurs publics ou privés [1]. Elle va devenir incontournable, du fait des besoins croissants en données pour l'apprentissage automatique, qui ne pourront systématiquement être assurés par un seul acteur. Au-delà des enjeux de concurrence et de régulation, la mutualisation de données représente un défi technique réel : faut-il mettre en commun les données des différents acteurs, et si oui à qui les confier ? Est-il possible au contraire de construire un modèle sans que les données ne quittent leur lieu de stockage respectif ? Dans le domaine de la défense, ces questions revêtent une importance particulière, du fait de la sensibilité des données manipulées.

Plusieurs techniques issues de la cryptographie et du calcul distribué peuvent alors être employées et éventuellement combinées avec la confidentialité différentielle. Nous les présentons dans la troisième partie avant de suggérer quelques cas d'usages envisageables pour la défense à différentes échelles.

2 Confidentialité différentielle pour l'apprentissage statistique

2.1 Principe

Le concept de confidentialité différentielle (*differential privacy*) a été introduit en 2006 [7], afin de protéger les individus fournissant des données à caractère personnel contre le risque de ré-identification. Considérons un algorithme randomisé $\mathcal{A}$ qui prend en entrée des données $\mathcal{D}$ et retourne une sortie aléatoire $\mathcal{A}(\mathcal{D})$ dans un espace E. $\mathcal{A}$ a pour confidentialité différentielle $\varepsilon > 0$ si pour tous jeux de données $\mathcal{D}$ et $\mathcal{D}'$ qui diffèrent d'au plus une donnée, et pour tout sous ensemble S de E :

$$\mathbb{P}(\mathcal{A}(\mathcal{D}) \in S) \leq e^{\varepsilon}\mathbb{P}(\mathcal{A}(\mathcal{D}') \in S)$$

Cela signifie que substituer une donnée à une autre n'affecte que très peu la sortie de l'algorithme. Plus ε est proche de 0 et plus la confidentialité est forte,

1. www.entreprises.gouv.fr/numerique/mutualisation-de-donnees-pour-intelligence-artificielle

$\varepsilon = 0$ signifiant que chaque donnée n'a aucune influence sur le résultat. Dans le cadre de l'apprentissage statistique, l'algorithme considéré est le processus d'entraînement dont la sortie est le modèle (on s'intéresse ici au contexte *white box*). Le compromis recherché est donc de construire un modèle utile, qui repose sur des informations du jeu d'entraînement, sans révéler trop d'informations sur chaque exemple particulier. Ce dilemme n'est qu'apparent, car le but de l'apprentissage est bien la généralisation, qui consiste à apprendre des caractéristiques générales d'une distribution de données et non les caractéristiques particulières de certaines données d'entraînement. Il s'agirait alors de sur-apprentissage, qui désigne une adaptation excessive au jeu de données d'entraînement. Néanmoins, les techniques employées actuellement pour apprendre de façon confidentielle conduisent empiriquement à une dégradation des performances sur le jeu de données de test.

Voici un exemple simple d'algorithme satisfaisant la confidentialité différentielle (ici $e^{\varepsilon} = 3$). Supposons que l'on cherche à estimer la proportion de consommateurs de drogues dans une population. Il est possible de poser directement la question à un échantillon de la population, cependant la réponse compromet la vie privée de chaque individu sondé. On peut proposer le processus suivant : lancer une pièce, si c'est pile, l'individu répond sincèrement ; sinon lancer une seconde pièce et répondre au hasard oui (face) ou non (pile). De cette façon, chaque individu peut réfuter sa réponse en prétendant qu'elle est due au hasard. Quant au sondeur, s'il dispose d'un échantillon assez large, il peut facilement retrouver une estimation fiable de la proportion de consommateurs de drogues à partir de la fréquence de réponses positives qu'il observe.

2.2 Propriétés et méthodes

Cet exemple met en lumière plusieurs propriétés fondamentales de la notion. La plus élémentaire est la robustesse face au post-traitement. Il n'est pas possible de compromettre davantage la vie privée d'un individu en « réfléchissant » à sa réponse. Une autre propriété importante est celle de la composition. Si deux requêtes de confidentialités respectives ε_1 et ε_2 sont effectuées sur les mêmes données, alors la composition des deux a une confidentialité $\varepsilon_1 + \varepsilon_2$. En d'autres termes, effectuer plusieurs requêtes sensibles dégrade significativement la protection envers les données. Cette propriété est intuitive : si l'on répète 100 fois le sondage décrit plus haut sur la même personne, on obtiendra une estimation très fiable de sa vraie réponse. Une dernière propriété utile est celle du sous-échantillonnage : si un individu a une probabilité inférieure à un d'être inclus dans l'étude, alors sa vie privée est davantage préservée.

Comment construire un algorithme qui satisfait la propriété de confidentialité différentielle ? Si l'algorithme donne une réponse déterministe qui dépend des données, c'est impossible. La solution est d'introduire un bruit aléatoire dans la réponse retournée. Dans l'exemple précédent, le principe utilisé est celui de la réponse randomisée : avec une certaine probabilité, une réponse aléatoire est renvoyée à la place de la sortie calculée. Lorsque l'algorithme renvoie une sortie dans un domaine continu, la méthode la plus courante est d'ajouter une

bruit aléatoire tiré selon une loi Gaussienne, en utilisant toutefois une définition légèrement relaxée de la confidentialité différentielle [6].

Dans le cas particulier de l'entraînement d'un modèle d'apprentissage statistique, il s'agit de garantir que l'algorithme d'optimisation utilisé pour minimiser l'erreur d'entraînement, typiquement un algorithme de descente de gradient stochastique, vérifie la confidentialité différentielle. Il s'agit probablement de l'application de la confidentialité différentielle qui a fait l'objet de la plus grande attention ces cinq dernières années. Plusieurs approches ont été étudiées :

— la perturbation de la sortie [5] : l'algorithme d'optimisation est appliqué normalement, puis un bruit Gaussien est ajouté au modèle final ;
— la perturbation de l'objectif [4] : la fonction objectif à minimiser est perturbée par l'ajout d'un bruit Gaussien ;
— la perturbation du gradient [1] : chaque gradient utilisé pendant les itérations de l'algorithme de gradient est perturbé par un bruit Gaussien ;
— l'agrégation bruitée de modèles entrainés sur des données disjointes [10].

La confidentialité différentielle est une garantie forte qui a la particularité de s'appliquer à un algorithme et non à un résultat. Ainsi, il n'est pas possible de certifier qu'un modèle vérifie cette propriété sans avoir accès à l'algorithme qui l'a construit. Cette garantie est fournie de façon constructive en utilisant notamment les propriétés de composition ou de sous-échantillonnage.

2.3 Intérêt pour la défense

Considérons par exemple la situation suivante : on cherche à détecter automatiquement tous les bâtiments dans une zone. Pour cela, on dispose de données protégées à différents niveaux, ainsi que de données non protégées (données cartographiques ouvertes ou extraites d'images commerciales), recueillies dans d'autres zones. Le modèle le plus performant utilisera l'intégralité des données comme un seul corpus. Il devra *a priori* être protégé au même niveau que la plus sensible des données utilisées, voire au-delà car on sait que le croisement de données peut en augmenter la sensibilité. Si l'on cherche au contraire à diffuser plus largement le modèle, la confidentialité différentielle permet de construire un second modèle, soit en ajoutant directement du bruit au premier, soit en le ré-entraînant en utilisant la méthode de perturbation du gradient. En pratique, le gain de confidentialité se fera bien souvent au prix de performances moindres.

La confidentialité différentielle est difficile à interpréter. Elle assure toutefois une protection efficace face aux attaques d'appartenance et de reconstruction [3], pour des paramètres de confidentialité ε raisonnables [9]. De plus, la définition usuelle correspond à un outil développé pour la protection de la vie privée, qui nécessite certaines adaptations. Elle garantit que chaque donnée individuelle a peu d'influence, mais pour certaines applications, c'est l'interaction entre plusieurs données individuelles qui peut s'avérer sensible. Une approche plus adaptée serait celle de la confidentialité de groupe (*group privacy*) [8], une extension naturelle de la confidentialité différentielle. Dans ce cas, on s'intéresse à la stabilité d'un algorithme face à la modification d'un groupe de données, par exemple de tous les membres d'une même famille. Concrètement, cette notion

pourrait être utilisée pour protéger les données issues d'une zone géographique, ou un ensemble de données provenant d'une même source.

Pour finir, la confidentialité différentielle ne garantit pas une confidentialité parfaite. Elle garantit seulement que la perte de confidentialité due à l'utilisation d'un algorithme n'est pas pire que si la donnée ou le groupe de données n'avait pas été utilisé. Dans une population homogène, la vie privée d'un individu qui n'a pas participé à une étude peut aussi en pâtir. Par exemple si une étude conclut à un haut niveau de criminalité dans une ville, alors tous les habitants de cette ville en seront impactés, qu'ils aient participé à l'étude ou non.

3 Cas des données distribuées

Il arrive que les données utilisées pour entraîner un modèle soient stockées sur différentes machines. C'est le cas des capteurs et autres objets connectés, qui disposent en général de faibles capacités de calcul. Une solution naturelle est de centraliser les données en les transmettant de façon sécurisée, par exemple en les chiffrant. Mais cela suppose de faire confiance à l'acteur central qui devient d'autant plus vulnérable qu'il possède une grande quantité de données. C'est cette centralisation excessive qui a récemment été reprochée aux géants du web.

Il est toutefois possible d'entraîner un modèle sans que les données ne quittent les machines qui les ont générées. C'est ce que propose l'apprentissage distribué [12], qui modifie le processus d'optimisation du modèle, de façon à ce que seuls les gradients, éventuellement compressés, de l'objectif à optimiser soient envoyés à un opérateur central. Cela peut présenter un avantage significatif en termes de coût de communication, notamment lorsque les capacités du réseau sont modestes ou dégradées. C'est l'occasion aussi de bruiter ces gradients pour assurer une certaine confidentialité différentielle. Dans certains cas, le processus d'apprentissage peut s'effectuer de façon purement décentralisée, par des communications locales entre nœuds adjacents d'un réseau (*gossip*) [11]. La sécurité du processus face à un adversaire surveillant les communications peut aussi être assurée par des protocoles de calcul multipartite sécurisé (*secure multi-party computation*) [13].

Certaines situations exigent au contraire d'utiliser les capacités de calcul d'un acteur centralisé, auquel on ne peut pas faire confiance. C'est le cas par exemple de l'utilisation des services de *cloud computing* commerciaux. Des protections sont fournies par la cryptographie, en particulier par le chiffrement totalement homomorphe (*fully homomorphic encryption*) [2], qui permet d'effectuer des opérations simples directement sur des messages chiffrés, au prix d'un ralentissement significatif.

Une question sous-jacente mérite d'être abordée : peut-on toujours faire confiance à l'acteur qui traite les données ? Elle se pose évidemment si l'acteur est un fournisseur de service externe à l'organisation, mais également dans le cas d'un *data scientist* interne. Il faut ainsi réfléchir à la façon dont le besoin d'en connaître pourrait être appliqué aux individus qui doivent traiter et explorer des données sensibles, sans pour autant entraver leur tâche. Certaines

organisations utilisant des données médicales appliquent des règles de limitation quotidienne du nombre de requêtes sensibles (*privacy budget*), l'impossibilité d'accéder à certaines informations insuffisamment agrégées, ou encore l'utilisation de requêtes pré-compilées pour éviter leur adaptation excessive à certaines données particulières.

4 Cas d'usages et architectures envisageables

Nous présentons ici quatre scénarios présentant un intérêt pour la défense, et qui nécessitent l'emploi d'une ou plusieurs techniques de protection des données d'entraînement.

4.1 Exportation de modèles entraînés avec des données classifiées

Imaginons la situation suivante (figure 1). Un État cherche à construire un système de reconnaissance automatique de véhicules sur des images aériennes captées par un drone. Ce développement est confié à un industriel, qui dispose de plusieurs jeux de données d'entraînement :

- des données disponibles en *open source* et en grande quantité, mais peu spécialisées, par exemple contenant uniquement des véhicules civils ;
- un jeu de données spécifiques au problème, construit au préalable par l'industriel et lui appartenant ;
- un jeu de données classifiées, fourni spécifiquement pour l'étude, afin de couvrir des cas d'usages importants et difficiles à traiter autrement.

Le système entraîné est destiné à être intégré à un programme d'armement, le drone, qui pourra également faire l'objet d'une exportation. Dans ce cas se pose la question des contributions respectives des trois jeux de données d'entraînement à la performance du modèle final, ainsi qu'aux potentielles fuites de données sensibles. L'usage de la confidentialité différentielle de groupe permet de moduler finement les contributions de chacun des jeux de données.

Ainsi, l'industriel pourra produire deux versions du système de reconnaissance : le modèle θ_1 entraîné sur toutes les données, sans confidentialité différentielle, et le modèle θ_2 destiné à l'export. Celui-ci aura un niveau de confidentialité ε qui pourra être débattu contractuellement, ainsi qu'une performance $p(\varepsilon)$ moindre pour des niveaux de confidentialité élevés.

4.2 Coopération sans partage de données

Plusieurs États coopèrent pour entraîner conjointement un modèle de classification de signaux électromagnétiques (figure 2). Compte-tenu de la sensibilité de leurs jeux de données respectifs, ils ne souhaitent pas les partager.

Ils peuvent dans ce cas utiliser le chiffrement totalement homomorphe. Les données sont transmises chiffrées à un prestataire qui effectue des opérations directement sur ces données chiffrées. Celui-ci n'a jamais accès aux données

en clair, ni au véritable modèle entraîné. Celui-ci est produit chiffré, puis renvoyé à chacunes des parties qui possède une clé pour le déchiffrer. Cela présente néanmoins l'inconvénient de rendre le processus d'apprentissage considérablement plus lent que s'il était effectué en clair, ce qui limite en pratique son usage à grande échelle.

Il faut noter que cette technique n'empêche pas d'éventuelles fuites d'informations contenues par les données d'entraînement par le modèle final. C'est pourquoi il peut être judicieux de la combiner avec la confidentialité différentielle.

4.3 Apprentissage conjoint centralisé

On cherche à construire un système de classification automatique de signatures acoustiques, à partir de données recueillies par plusieurs sous-marins (figure 3). Leurs capacités de transmission sont limitées, mais ils disposent de capacités de calculs propres.

L'apprentissage fédéré consiste à mettre en commun, non pas directement les données d'entraînement, mais les modèles entraînés sur chaque sous-ensemble de données, pour construire un modèle conjoint plus performant. Il est pertinent lorsque les canaux de communication ne permettent pas de transmettre l'ensemble des données, mais que les modèles intermédiaires, plus compacts, peuvent l'être. Le modèle conjoint est ensuite redistribué à chacun des contributeurs.

4.4 Apprentissage conjoint décentralisé

Cette situation (figure 4) est une variante de la précédente. Les données sont disponibles en différents nœuds d'un réseau déployé, représentant par exemple différents groupes de combat ou des bases. Le réseau n'est pas centralisé, et chaque appareil ne peut communiquer qu'avec ses voisins.

Un modèle commun peut être entraîné grâce à des techniques d'apprentissage décentralisé, en n'utilisant que des communications locales. Ce processus est itératif et nécessite plusieurs étapes alternant communications et calculs effectués localement. Il présente l'avantage d'être résilient à la rupture de certaines communications et évite de centraliser les données de façon excessive, ce qui exposerait le nœud central à des attaques.

Même si les cas d'usages dans le domaine de la défense sont peu nombreux aujourd'hui, ils sont amenés à se développer dans le civil, notamment avec la diffusion de la technologie 5G.

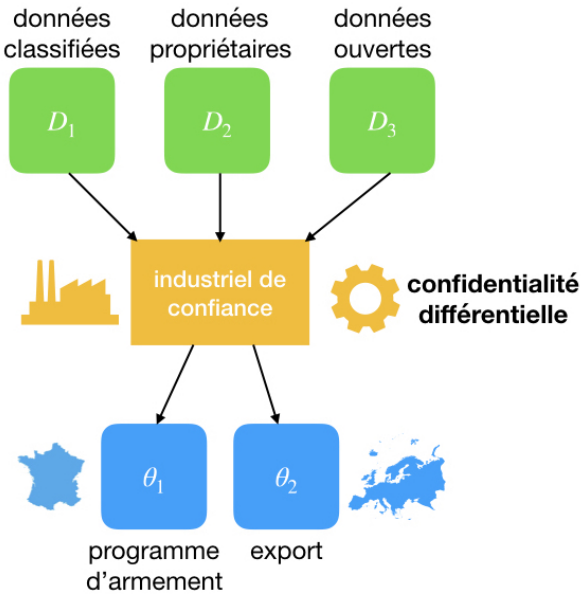

Figure 1. Exportation de modèles entraînés avec des données classifiées

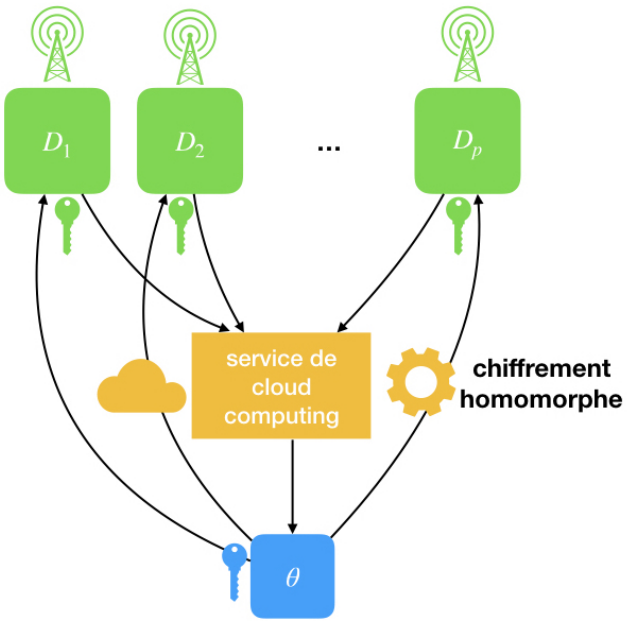

Figure 2. Coopération sans partage de données

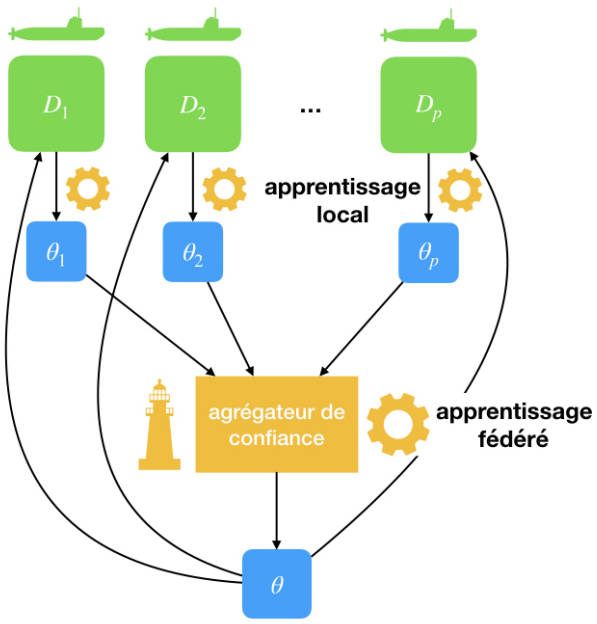

Figure 3. Apprentissage conjoint centralisé

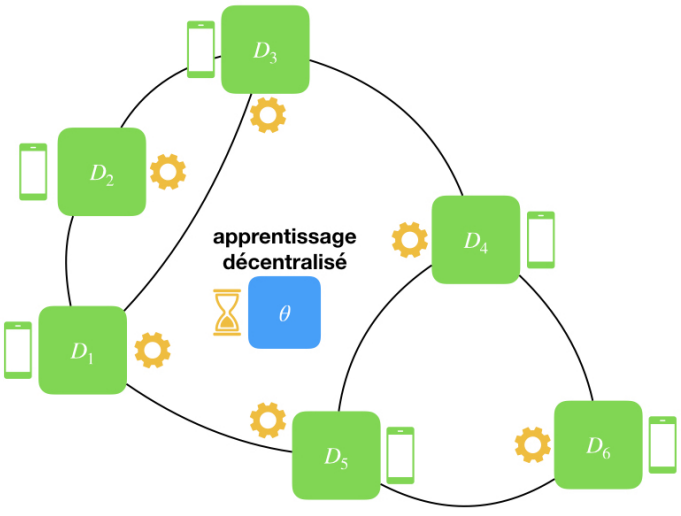

Figure 4. Apprentissage conjoint décentralisé

5 Conclusion

Les données, en partie classifiées, produites dans la sphère défense vont être de plus en plus sollicitées, notamment pour des usages liés à l'apprentissage automatique. La protection du secret devra donc s'adapter à ces nouveaux usages et mettre en application de nouvelles méthodes pour protéger les données sans pour autant les dissimuler. Il s'agira de trouver un équilibre entre les opportunités offertes par l'exploitation automatique des données et leur compromission.

Il convient de séparer la confidentialité du processus d'entraînement d'un modèle de celle du modèle lui-même. D'une part, le processus d'entraînement, parce qu'il nécessite de déplacer et d'échanger des données, est susceptible de les exposer. Les techniques de chiffrement homomorphe ou d'apprentissage distribué permettent de sécuriser le processus d'apprentissage. D'autre part, le modèle entraîné contient lui-même des informations sensibles sur ses données d'entraînement. Il faut en être conscient et préconiser par exemple l'usage de la confidentialité différentielle lorsqu'une diffusion plus large du modèle est envisagée.

Références

1. Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L. : Deep learning with differential privacy. In : Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 308–318. ACM (2016)
2. Bost, R., Popa, R.A., Tu, S., Goldwasser, S. : Machine learning classification over encrypted data. In : NDSS. vol. 4324, p. 4325 (2015)
3. Carlini, N., Liu, C., Kos, J., Erlingsson, Ú., Song, D. : The secret sharer : Measuring unintended neural network memorization & extracting secrets. arXiv preprint arXiv :1802.08232 (2018)
4. Chaudhuri, K., Monteleoni, C., Sarwate, A.D. : Differentially private empirical risk minimization. Journal of Machine Learning Research **12**(Mar), 1069–1109 (2011)
5. Chen, C., Lee, J., Kifer, D. : Renyi differentially private erm for smooth objectives. In : The 22nd International Conference on Artificial Intelligence and Statistics. pp. 2037–2046 (2019)
6. Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M. : Our data, ourselves : Privacy via distributed noise generation. In : Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 486–503. Springer (2006)
7. Dwork, C., McSherry, F., Nissim, K., Smith, A. : Calibrating noise to sensitivity in private data analysis. In : Theory of cryptography conference. pp. 265–284. Springer (2006)
8. Dwork, C., Roth, A. : The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science **9**(3–4), 211–407 (2014)
9. Jayaraman, B., Evans, D. : When relaxations go bad :" differentially-private" machine learning. arXiv preprint arXiv :1902.08874 (2019)

10. Papernot, N., Song, S., Mironov, I., Raghunathan, A., Talwar, K., Erlingsson, Ú. : Scalable private learning with pate. arXiv preprint arXiv :1802.08908 (2018)
11. Scaman, K., Bach, F., Bubeck, S., Massoulié, L., Lee, Y.T. : Optimal algorithms for non-smooth distributed optimization in networks. In : Advances in Neural Information Processing Systems. pp. 2740–2749 (2018)
12. Smith, V., Forte, S., Chenxin, M., Takáč, M., Jordan, M.I., Jaggi, M. : Cocoa : A general framework for communication-efficient distributed optimization. Journal of Machine Learning Research **18**, 230 (2018)
13. Yao, A.C.C. : How to generate and exchange secrets. In : 27th Annual Symposium on Foundations of Computer Science (sfcs 1986). pp. 162–167. IEEE (1986)