SECMPNN: 3-PARTY PRIVACY-PRESERVING MOLECULAR STRUCTURE PROPERTIES INFERENCE

Xinying Liao[1] *Jiaye Xue*[1] *Shengxing Yu* [2,*] *Ximeng Liu*[1,3] *Jiangang Shu* [3]

[1] College of Computer and Data Science, Fuzhou University, Fuzhou 350108, China
[2] School of Electronics Engineering and Computer Science, Peking University, China
[3] Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen

ABSTRACT

Compound screening is a key step in the development of new drugs. Current high-throughput screening methods cannot be widely adopted by laboratories due to their expensive equipment and low efficiency. The booming deep learning in recent years has provided a new answer to this question. The message passing neural network (MPNN) can directly predict molecular properties from molecular structure so that compound screening can be completed without experimentation. In the face of large-scale molecular data, outsourcing this task to a professional cloud server can further accelerate prediction efficiency and reduce costs. In order to solve the privacy protection problem of computing on cloud servers, we propose a 3-party molecular structure properties inference privacy protection framework SecMPNN based on additive secret sharing. We design brand-new cryptographic protocols to ensure the privacy and security in the prediction process, and through experiments show that the single inference time of our protocol on different networks is 40.85% faster than CRYPTEN and 18.5% faster than SecureNN.

***Index Terms*—** Privacy-preserving, Secret Sharing, MPNN, Molecular Structure Properties Inference

1. INTRODUCTION

According to the survey [1, 2], the average time required for a new drug from research and development to market launch is 13.5 years, and the average cost paid by pharmaceutical companies is $1.395 billion. The entire research and discovery cycle can be divided into preclinical research and clinical research. The preclinical research mainly takes about 3∼6 years to complete the screening of compounds. However, the current mainstream high-throughput screening method cannot be used by most laboratories due to its expensive instrument requirements and unsatisfactory efficiency. With the continuous development of deep learning technologies, it has new applications in the field of compound screening. The message passing neural network (MPNN) [3] can inference the properties of the compound through its molecular structure, and rapid preliminary screening of the compound without the aid of experiments can greatly reduce compound screening time. Because of the huge number of compounds that need to be predicted, instead of building its own local server, the drug laboratory might as well outsource this computing task to a cloud server to further reduce costs.

However, there are serious privacy concerns in practical application [4]. The drug laboratory is unwilling to disclose the drug data and trained model parameters to the cloud server [5]. To address such privacy concerns, several recent works [6–9] have proposed cryptographic protocols to ensure privacy in inferences base on Garbled Circuits (GC) [10], Homomorphic Encryption (HE) [11], Differential Privacy (DP) [12] and Secure Multiparty Computation (SMC) [13]. For example, CRYPTEN [14] proposed a practically efficient framework in the two-server model using 2-party protocols that perform computation in arithmetic and binary. SecureNN [15] implements a maliciously secure 3-party SMC protocol. The AriaNN [16] framework considered a semi-honest 2-party setting where protocol leverages secret sharing. 2-partys framework ABY [17] and its extension to 3-partys ABY3 [18] using three different data forms.These frameworks often rely on expensive cryptographic protocols, which require very high computing power and huge traffic. The data available in drug laboratory are huge and grow rapidly. In the face of such a practical situation, these frameworks can not be applied to such a practical situation. Thus, we are seeking an efficient scheme to securely implement the MPNN inferences in the cloud.

In order to solve the above problem, we propose SecMPNN, an efficient and security 3-party molecular structure properties inference framework by co-designing additive secret sharing, deep learning and cloud computing technologies. We propose division, comparison, and exponential protocols to complete secure calculations on cloud servers. And through experiments, it can be shown that the execution time of the comparison protocol we designed is faster than CRYPTEN by

Thanks to the Key-Area Research and Development Program of Guangdong Province under Grant No.2020B0101360001, the National Natural Science Foundation of China (No.62072109, No.U1804263, No.62102204) and Natural Science Foundation of Fujian Province (No. 2021J06013). * Corresponding author.

more than 2× and 3× faster than SecureNN, and the execution speed of other protocols has also been improved to varying degrees. Existing solutions mostly set the data encryption length to a fixed bit length. In actual situations, there may be cases where the encryption length is too short and not secure enough. We set security parameters to make the encryption length selectable to make SecMPNN more secure.

2. PROPOSED FRAMEWORK

In this section, we describe the system model and security model of SecMPNN and the process of using SecMPNN to make inference.

2.1. System Model

As illustrated in Fig.1, SecMPNN comprises by two types of participants, namely drug laboratory DL, three cloud servers $S = \{S_0, S_1, S_2\}$.

- DL is drug laboratory, which would like to use the computational power of cloud server to inference. DL is not willing to share their drug data and model parameters to cloud server. Therefore, in SecMPNN, DL randomly split the drug data into $D = \{D_0, D_1, D_2\}$ and send to the corresponding cloud servers for computation.

- S_0, S_1 and S_2 are three outsourced cloud servers which are responsible for the intensive computations task. In SecMPNN, they complete all the computations of MPNN without knowing any plaintext of drug data. The final outputs O_0, O_1, O_2 are simultaneously sent to DL. The drug laboratory can obtain the original output by computing $O = O_0 + O_1 + O_2$.

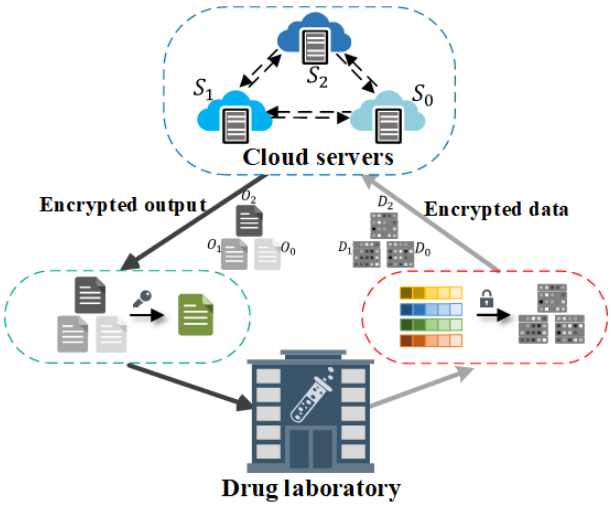

Fig. 1. System Model

2.2. Security Model

In SecMPNN, we use the semi-honest (honest-but-curious) security model. That is, cloud servers will perform the protocols exactly as specified, yet may attempt to learn as much as possible about private information of interest-based one the data stored and processed on it. Three cloud servers S_0, S_1, S_2 are assumed to be independent and non-colluding. It means that one cloud server will not reveal more information than protocol messages to the other.

2.3. Inference Procedure of SecMPNN

SecMPNN is implemented based on MPNN. MPNN is a graph neural network. This network consists of two stages, the message passing stage and the read-out stage. The message passing stage contains two main functions, message function and update function. The realization of the message function is to collect the information of all neighbors connected to the current point by constructing a neural network including fully connected layers and ReLU functions. In SecMPNN, this function is SecMe. The update function is actually the GRU network [19], the collected information is updated to the current point by update gate and reset gate. In SecMPNN, this function is SecUp. The final readout stage is to calculate the feature vector of the entire graph. In SecMPNN, the function is SecRe. Fig.2 shows the detailed structure of the three functions.

3. SUPPORTING PROTOCOL

3.1. Protocol Overview

In order to complete the inference process, a secure division is designed based on the principle of continuously decomposing the dividend into specified multiples of the divisor. If the dividend is less than the divisor, the dividend is expanded and then decomposed until the specified accuracy is reached. The design of the secure exponential is mainly to carry out the process of converting from multiplicative sharing to additive sharing. Because in the case of an exponential, each party holds a part of the power, and the three parties multiply to get the final result, so it needs to be converted to additive. Although these protocols are designed in a 3-party scenarios, our design is scalable and can be extended to n-party scenarios. The data types in the protocol include arithmetic and binary and the range of data is in Z_{2^n}. By changing n, We can provide more security through this security parameter.

3.2. Secure Comparison Protocol

This function evaluates $a >= b$ where the parties hold shares of a and b in $\mathbb{Z}_{2^n}$. Algorithm 1 describes this protocol. Servers gets the shares of 1 when $a >= b$, otherwise servers gets the shares of 0.

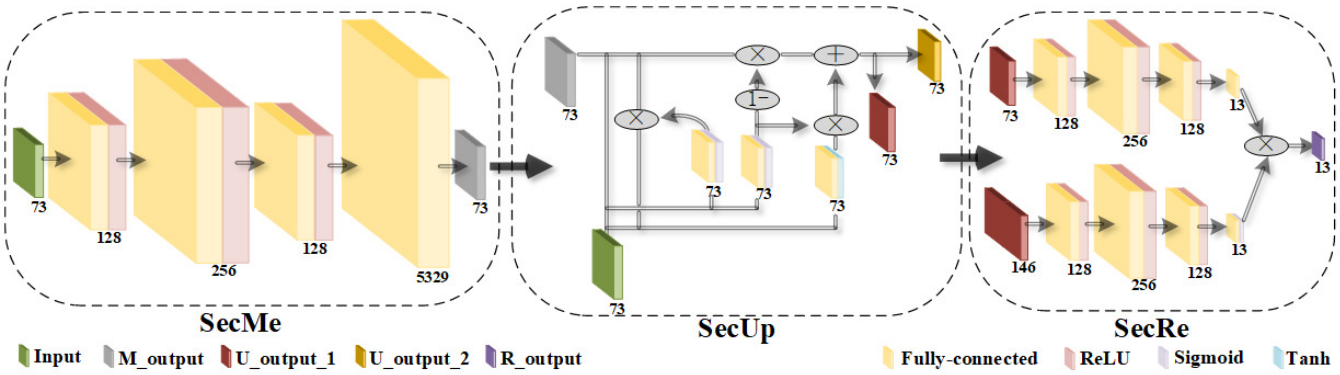

Fig. 2. Inference Procedure of SecMPNN

Algorithm 1 Secure Comparison Protocol

Input: Shared value $[a]$ and $[b]$.
Output: Shared value $[ans]$ where $ans = 1$ if $a >= b$ and $ans = 0$ otherwish.

1: Server S_0 generates shares of random number $[\alpha] \in \{0, 1\}$ and $[r] \in Z_{2^n}$, then sends α_1, r_1 to S_1 and sends α_2, r_2 to S_2
2: Compute $[c] = [\alpha] \times ([a] + 1 - [b]) + (1 - [\alpha]) \times ([b] - [a])$, then $[d] = [c] \times [r]$
3: S_0, S_1 send d_0, d_1 to S_2. S_2 compute the sum $e = d_0 + d_1 + d_2$.
4: S_2 generates shares of 1 and 0, then sends the shares to each servers. if $e > 0$, $[ans] = [\alpha] \oplus 0$, otherwise $[ans] = [\alpha] \oplus 1$
5: **return** $[ans]$

Step 1: Server S_0 generates shares of random number $[\alpha] = 1$ or $[\alpha] = 0$, this index is used for selection. $[r]$ is used to hide value.

Step 2: Using the random number generated in the Step 1 to calculate $[c] = [\alpha] \times ([a] + 1 - [b]) + (1 - [\alpha]) \times ([b] - [a])$. In fact, this formula uses random numbers to select two calculations of $a + 1 - b$ and $b - a$, in order not to disclose information when revealing the results. Then use the $[r]$ to hide the value of $[c]$.

Step 3: S_0, S_1 send d_0, d_1 to S_2, then S_2 calculate $[e]$ the sum of c_i.

Step 4: Determine the final value according to whether $e > 0$. if the $e > 0$, $[\alpha] \oplus 0$ will be the final answer, otherwise, the answer is $[\alpha] \oplus 1$.

Security analysis. Each server can only get the shared value of the input value, which can ensure the security of the input data. In the calculation process of the comparison protocol, the server S_0 needs to generate α and r, but these two values are only random numbers used for selection and hiding. S_0 and S_1 can only get the shared value in the calculation process and cannot get any useful information. The server S_2 can get the value hidden by r. The sign of this value does not directly get the final result, and the final result is related to α. In our security model, it is stipulated that no collusion between any two servers is allowed, so the security of the final result can be guaranteed.

	Divison	Exponential	Comparison
CRYPTEN	32.4ms	12.1ms	8.33ms
SecureNN	56ms	8.6ms	14.1ms
AriaNN	80.6ms	13.5ms	6.27ms
Our	24.8ms	6.29ms	4.58ms

Table 1. Comparison Time of The Sub-protocols

4. EXPERIMENTAL EVALUATION

Our experiments implemented on the server which has sixteen 64-bit Intel(R) Xeon(R) Gold 5218 core with 2.3GHZ, 128G memory installed with Ubuntu 16.04. Three datasets are used in our experiment:(1) MNIST dataset [20] includes 70k 28×28 handwritten images; (2) CIFAR10 dataset [21] includes 60k 32×32 colour images in 10 class. (3) QM9 dataset [22] includes 134k stable small organic molecules made up of CHONF. In our experiments, four types network model has been used.

Network A consists of fully connected layers and activation functions, and the network uses the MNIST dataset; network B is a LeNet5 [20] network with CIFAR10 as the dataset, which mainly includes fully connected layers, pooling layers, ReLU functions and convolutional layers; Network C also uses the MNIST dataset, including fully connected layers, pooling layers, ReLU functions, and convolutional layers; MPNN is a graph neural network consisting of three small networks consisting of fully connected layers and activation functions and a GRU network.

	Network A	Network B	Network C
CRYPTEN	0.69s	7.81s	12.43s
SecureNN	0.42s	5.67s	10.39s
AriaNN	0.27s	4.81s	9.84s
Our	0.11s	4.62s	7.34s

Table 2. Single Inference Time of Different Networks

4.1. Performance Evaluation of Sub-protocols

Secure comparison is a fundamental building block, necessary for the realization of various protocols: secure division, secure exponential, secure activation function, etc. Most recent articles complete the secure comparison by converting arithmetic numbers into binary form. Most of this conversion process relies on complex and inefficient bit extraction protocols. The comparison protocol we proposed does not require this inefficient conversion process, and completes the secure comparison operation directly in arithmetic form. This contribution has greatly improved the efficiency of secure comparison. In the experiment, we compared the three functions of division, exponential and comparison, and tested the time to perform a single operation for protocols of different frameworks under the same data. It can be seen from Table 1 that our comparison protocol has the best efficiency, which is at least 26.9% higher than other frameworks.

4.2. Performance Evaluation in Different Network

This experiment compares the inference time of our proposed cryptographic protocols and other frameworks such as CRYPTEN, SecureNN, and AriaNN in different networks. From the results in Table 2, we can see that in the inference of network A, the inference completed by our protocol are the fastest, 6× faster than CRYPTEN, 4× faster than SecureNN, and 2× faster than AriaNN. Compared with CRYPTEN, the inference achieved by our protocol in network B has an improvement of 40.85% and an 18.5% improvement compared to the inference time of SecureNN. Compared with CRYPTEN, SecureNN and AiraNN, our protocol has 40.9%, 29.35%, and 25.4% improvement in single inference time in network C.

In the experiment of different network single inference time, we also recorded the correct rate of the protocol we designed in different network inference. It can be seen from Table 3 that for network A, the accuracy of the security inference using our protocol is only 0.11% lower than the accuracy of the inference under the plaintext. For network B, the inference accuracy rate of our protocol is the same as the accuracy rate under the plaintext. For network C, the accuracy of the inference using our protocol is only 0.02% lower than the accuracy under the plaintext. For the more complex MPNN network, the security inference using our protocol is only 1.52% lower than that in the plaintext. The calculation error mainly comes from the truncation error caused by the same data format as the integer.

	Training Accuracy	Inference Accuracy	Our Protocol Inference Accuracy
Network A	94.96%	94.23%	94.12%
Network B	97.81%	95.50%	95.50%
Network C	96.16%	80.02%	80%
MPNN	98.17%	97.93%	96.42%

Table 3. Inference Accuracy of our protocol in Different Network

4.3. Performance Evaluation of SecMPNN

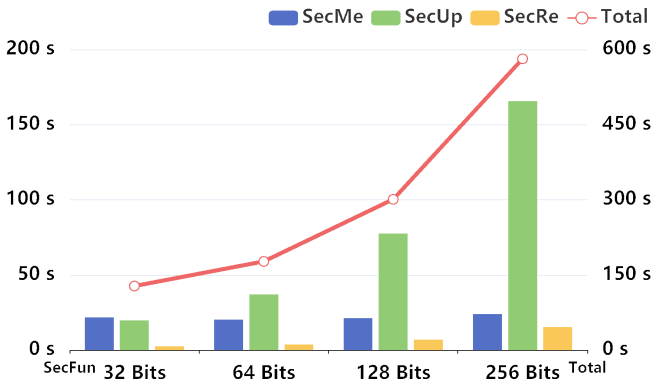

Fig. 3. Runtime of SecMPNN

In this experiment, we tested the different phases and total time of molecular structure properties inference using the MPNN network under the QM9 dataset in the case of four different encryption lengths of 32 bits, 64 bits, 128 bits and 256 bits. It can be seen from Fig.3 that the total time increases as the encryption length increases, and it takes about two minutes to predict one molecule under the encryption length closer to the actual 32 bits encryption length. For the three phases of SecMPNN, the time of the SecUp phase is most obviously affected by the encryption length and the time spent is the longest. This is because the protocol used in SecUp involves binary data types, which will increase with the encryption length and consume more time.

5. CONCLUSION

We proposed a privacy protection framework SecMPNN for molecular structure attribute prediction. Because there is no need to perform inefficient arithmetic to binary conversion, the efficiency of our comparison protocol has been greatly improved and data security can still be guaranteed. This has also improved the efficiency of other comparison-based protocols, which has led to a better performance of SecMPNN.

6. REFERENCES

[1] Joseph A DiMasi, Henry G Grabowski, and Ronald W Hansen, "Innovation in the pharmaceutical industry: new estimates of r&d costs," *Journal of health economics*, vol. 47, pp. 20–33, 2016.

[2] Steven M Paul, Daniel S Mytelka, Christopher T Dunwiddie, Charles C Persinger, Bernard H Munos, Stacy R Lindborg, and Aaron L Schacht, "How to improve r&d productivity: the pharmaceutical industry's grand challenge," *Nature reviews Drug discovery*, vol. 9, no. 3, pp. 203–214, 2010.

[3] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl, "Neural message passing for quantum chemistry," in *International conference on machine learning*. PMLR, 2017, pp. 1263–1272.

[4] Wenyun Dai, Meikang Qiu, Longfei Qiu, Longbin Chen, and Ana Wu, "Who moved my data? privacy protection in smartphones," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 20–25, 2017.

[5] Cheng Bo, Guobin Shen, Jie Liu, Xiang-Yang Li, Yong-Guang Zhang, and Feng Zhao, "Privacy. tag: Privacy concern expressed and respected," in *Proceedings of the 12th ACM conference on embedded network sensor systems*, 2014, pp. 163–176.

[6] Chiraag Juvekar, Vinod Vaikuntanathan, and Anantha Chandrakasan, "{GAZELLE}: A low latency framework for secure neural network inference," in *27th {USENIX} Security Symposium ({USENIX} Security 18)*, 2018, pp. 1651–1669.

[7] Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing, "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," in *International conference on machine learning*. PMLR, 2016, pp. 201–210.

[8] Peichen Xie, Bingzhe Wu, and Guangyu Sun, "Bayhenn: Combining bayesian deep learning and homomorphic encryption for secure dnn inference," *arXiv preprint arXiv:1906.00639*, 2019.

[9] Qiao Zhang, Cong Wang, Hongyi Wu, Chunsheng Xin, and Tran V Phuong, "Gelu-net: A globally encrypted, locally unencrypted deep neural network for privacy-preserved learning.," in *IJCAI*, 2018, pp. 3933–3939.

[10] Andrew Chi-Chih Yao, "How to generate and exchange secrets," in *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*. IEEE, 1986, pp. 162–167.

[11] Craig Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the forty-first annual ACM symposium on Theory of computing*, 2009, pp. 169–178.

[12] Chengkun Wei, Shouling Ji, Changchang Liu, Wenzhi Chen, and Ting Wang, "Asgldp: collecting and generating decentralized attributed graphs with local differential privacy," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3239–3254, 2020.

[13] Donald Beaver, "Efficient multiparty protocols using circuit randomization," in *Annual International Cryptology Conference*. Springer, 1991, pp. 420–432.

[14] Brian Knott, Shobha Venkataraman, Awni Hannun, Shubho Sengupta, Mark Ibrahim, and Laurens van der Maaten, "Crypten: Secure multi-party computation meets machine learning," *arXiv preprint arXiv:2109.00984*, 2021.

[15] Sameer Wagh, Divya Gupta, and Nishanth Chandran, "Securenn: 3-party secure computation for neural network training.," *Proc. Priv. Enhancing Technol.*, vol. 2019, no. 3, pp. 26–49, 2019.

[16] Théo Ryffel, Pierre Tholoniat, David Pointcheval, and Francis Bach, "Ariann: Low-interaction privacy-preserving deep learning via function secret sharing," *arXiv preprint arXiv:2006.04593*, 2020.

[17] Daniel Demmler, Thomas Schneider, and Michael Zohner, "Aby-a framework for efficient mixed-protocol secure two-party computation.," in *NDSS*, 2015.

[18] Payman Mohassel and Peter Rindal, "Aby3: A mixed protocol framework for machine learning," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 35–52.

[19] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[20] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[21] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton, "The cifar-10 dataset," *online: http://www. cs. toronto. edu/kriz/cifar. html*, vol. 55, no. 5, 2014.

[22] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld, "Quantum chemistry structures and properties of 134 kilo molecules," *Scientific data*, vol. 1, no. 1, pp. 1–7, 2014.