International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V70I6P206

Volume 70 Issue 6, 47-53, June 2022
© 2022 Seventh Sense Research Group®

Original Article

Machine Learning based Encryption Framework for Privacy Preservation in Covid 19 Data

R. Karunia krishnapriya[1], G. Vinodhini[2], R. Suban[3], K. Sakthivel[4]

[1,2]Department of CSE, Annamalai University, Chidambaram, India
[3]Department of Information Technology, Annamalai University, Chidambaram, India
[4]Department of CSE, K.S. Rangasamy College of Technology, India.

[1]sakthikkp@gmail.com

Received: 20 March 2022 Revised: 14 May 2022 Accepted: 26 May 2022 Published: 27 June 2022

Abstract - *With the ongoing pandemic of COVID-19, numerous bits of intelligence are required to analyse the type of infection and mutation that is undergoing around the globe. However, while processing these data, privacy is another major field that requires concentration since it involves user data privacy. Hence, preserving privacy while processing the images is required for optimal management and security consideration of big data analytics. In this paper, the study models a machine learning method that is integrated with encryption models for optimal data encryption and thereby maintaining the privacy preservation of data. The privacy preservation using the proposed data encryption model is tested under two problems involving digit recognition and medical image application in classifying the computed tomography images of lungs from various covid-19 patients. The simulation was conducted to test the efficacy of the privacy preservation method in providing data privacy while the data has been classified. The simulation results show that the proposed machine learning encryption model enables optimal privacy of user data than other existing methods.*

Keywords - *Privacy Preservation, Bigdata Analytics, Covid-19, Machine Learning.*

1. Introduction

With the closed training approach, machine learning has evolved into an intelligent and complex system. The operation in a cloud environment makes it possible to leverage massive amounts of data and computing power for large-scale data processing [1] [2]. At the same time, cloud computing enables artificial intelligence to be convenient, where its security issues are still a major concern. Using cloud storage means that customers will lose the ability to store machine learning data in a physically isolated location and protect their data [3].

It is also common practice to outsource some of the more complicated computations for artificial intelligence techniques and machine learning (ML) onto the servers in a cloud environment, which can result in illegal intrusion or data loss, as well as a host of other security problems [4]. Consequently, the preservation of privacy in the cloud environment has become a critical aspect of the advancement of machine learning [5] [6] [24] [25].

Homomorphic encryption theory has increased research into preserving machine learning data privacy [7] [8]. The work uses a support vector machine (SVM) classification to demonstrate ML based on the Homomorphic encryption technique. Using a cloud server, a user can develop a machine learning classifier without revealing the sensitive data using a huge data source and the cloud server assistance (see Fig. 1)

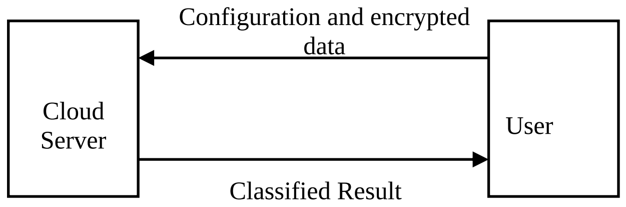

Fig. 1 Privacy Preserved Machine Learning

Model parameters are randomly initialized, and relevant configuration is encrypted using a specific model when the user interacts. The user then sends the ciphertexts and any relevant configuration information to the cloud server. As a final step, the server uses model parameters to classify with the help of ciphertexts. Then it returns the label associated with the users without disclosing their private data.

In this paper, the machine learning model is built by integrating the encryption process models for encryption and decryption, thereby preserving the covid-19data privacy.

The main support of the paper involves the subsequent:

- The authors developed a machine learning model to preserve the privacy of covid-19 data uploaded from various devices to a common cloud center.
- The privacy preservation using the machine learning-based data encryption model is tested on the feeding of privacy preserved data inputs digit recognition system and medical imaging data classification for lungs in relation with covid-19 patients.
- The simulation was conducted to test the efficacy of the privacy preservation method in providing data privacy while the data has been classified.

2. Related works

For the past ten years, ML algorithms have been built using homomorphic encryption. Based on homomorphism, encryption utilizing ElGamal multiplicative principle was developed by Chen et al. [9]. This study developed an algorithm for two-party distributed back propagation in multi-layer neural networks. It is possible to train neural networks without revealing encrypted data.

Homomorphic encryption was used with gradient descent algorithms for classification. [10] Paillier homomorphic encryption was used to secure a new classification strategy for users and servers presented by Mathavan [11] based on SVM.

Bost et al. [12] used to create certain classification protocols, such as hyperplane judgment and Naive Bayes, using additive homomorphism encryption. An asymmetric ciphertext-decryption method that allows the calculation of the nearest-neighbor algorithm was devised by Wong et al. [13] In the same year. Three algorithms were developed by Aslett et al. [14] to classify FHE ciphertexts using an FHE scheme: Naive Bayes, Complete Random Forest (CRF), and Logical Regression (LR).

According to Liu et al. [15], additive homomorphism uses a classifier called Naive Bayes to preserve sensitive data. It successfully estimates the risk of patient disease, according to Liu et al. [15]. Further, Dowlin et al. [16] developed an approximation machine learning model that uses neural network classifiers, and the encryption is conducted using CryptoNets with 99% accuracy.

Baryalai et al. [26] developed a dual cloud model to decentralise cloud power while improving the security of neural network systems. The authors in [18] developed SVM classifiers based on homomorphic encryption. Phong et al. [19] constructed a privacy protection system using deep learning to protect sensitive information by applying additive homomorphism encryption.

Homomorphic encryption is further developed over real numbers in [20, 21]. The authors in [20] created an approach using logic regression with a parallelization homomorphic encryption model. The homomorphic ciphertext is further developed in [22]. A secret key is required to further the decryption of intermediate ciphertext outcomes.

3. Proposed Method

Homomorphic encryption is performed on the input data in this section. For real-number encryption purposes, a matrix ring was proposed for homomorphic encryption. Finally, the study ensures that ML classification in COVID-19 input data is protected from privacy violations.

3.1. Homomorphic Encryption

Multiple users can benefit from homomorphic encryption to foresee the computations aggregating sensitive data. An extra attribute, dubbed threshold-HE in the study, is highly recommended for shared settings with numerous users. Interactive processes are used for the key generation and decryption algorithms, in which the participants work together to generate and decrypt keys.

3.2. Definition and Notations

ParamGen(λ, P, K, B) are considered the parameters. Parameters are generated using the parameter generation algorithm, which is explained below. It requires the following as input:

λ is the intended level of security for the mechanisms. For example, 128bits or 256-bit security (λ =128). A plaintext number modulus, P, is the number one that wants to encrypt. It is assumed that P = 1024 means that all operations on individual message elements are performed in the range (0, 1023) modulo P.

The encoding vectors have a dimension of K. For example, $K = 100, P = 1024$ means that the messages sensitive to the users are encrypted of vector type $(V_1, \dots, V_K)$ whereas all the vectors V_i is always selected between the range [0, 1023], and functions were conducted mostly in the order of components. According to the definition $(V_1, \dots, V_K) + (V_1', \dots, V_K') = (V_1 + V_1', \dots, V_K + V_K')$. Multiplying a pair of vectors is defined the same way. The message space refers to the set of all conceivable vectors $(V_1, \dots, V_K)$.

The study uses an auxiliary parameter, say B, which is useful in controlling the complexity associated with communication and computation of encryptions. As a rule of thumb, the lower the parameters, the smaller the program or circuit is. The smaller the scheme parameters, the more likely it is to have fewer parameters. As a result, the ciphertexts are smaller, and the assessment procedures are more efficient. The more parameters you use, the larger your keys and ciphertext will be, and the more difficult it will be to evaluate them. More complicated programs necessitate higher parameters for evaluation.

- PubKeygen(Params) → SK, PK, EK

Secret and public keys are generated using a private key generation technique. Anyone can encrypt messages using the public key. For decrypting encrypted messages, a user must access the confidential key. An evaluation key is further generated for the execution of operations of homomorphic encryption conducted on ciphertexts generated by the method itself. Entities performing homomorphic computations on the ciphertexts deserve this key. It is impossible to decipher a message using only the ciphertexts if you only have access to the public and calculation keys.

- SecKeygen(Params) → SK, EK

A confidential key was generated using the secret key-generation algorithm. Messages are encrypted and decrypted using this secret key. To be safe, the user should not share it with anyone. It also provides an evaluation key required for homomorphic operations on the ciphertexts, which are also generated by this approach. The evaluation key is provided to an entity that optimally does the homomorphic encryption on the ciphertext. Ciphertexts can only be deciphered if the evaluation key is known to the recipient.

- PubEncrypt(PK, M) → C

Public encryption uses the public key and the sensitive user data or message *M* as inputs to the algorithm. C is considered the ciphertext that is generated, and to meet the security requirements, this algorithm must typically be randomised.

- SecEncrypt(SK, M) → C

Some message *M* in the message space, as well as the secret key, is fed into the algorithm for secret encryption. *C* is the ciphertext generated by the algorithm. This approach's security features require using random or pseudo-random coins.

- Decrypt(SK, C) → M

The ciphertext *C* and secret key *SK* are inputs to the decoding procedure. Sends a message to the recipient *M*. A unique FAIL symbol may also be generated if decoding the message encrypted *M* fails.

- EvalAdd(EK, Params, C1, C2) → C3.

The key required for evaluation EK and two C1 and C2 ciphertexts are used as input to EvalAdd, which generates a ciphertext C3.
This feature of EvalAdd states that if C1 and C2 encrypt the encrypted elements of plaintext M1 and M2, then the encrypted elements of plaintext M1+M2 should be encrypted by C3.

- EvalAddConst(Params, EK, C1, M2) → C3.

The evaluation key (EK), the system parameters (Parms), a ciphertext C1, and plaintext M2 are inputs to EvalAddConst, which generates ciphertext C3.
Since the plaintext element M1 is encrypted in C1, then C3 should be encrypted in C1, and then EvalAddConst says this is accurate.

- EvalMult(Params, EK, C1, C2) → C3.

Inputs mainly include C1 and C2, EK, and system parameters. On the other hand, the outputs include C3, which generates the estimation key EK.
Assuming that C1 and C2 are both encryptions of plaintext element M1 and that C3 is encrypted of multiplication factor M1*M2, EvalMult is valid.

- EvalMultConst(EK, Params, C1, M2) → C3.

EvalMultConstant was considered a random method that accepts as input Params, the EK, an encrypted cipher text C1, and an unencrypted plain text M2 and generates an encrypted cipher text C3.

- Refresh(flag, Params, EK, C1) → C2.

Bootstraps are all flags that can be used as inputs to Refresh. The EK, C1, and C2 are all outputs of Refresh. C1 is plaintext encryption of M1 when C2 is accurate as per the correctness property.

Using the Refresh technique, a "complicated" ciphertext of messages can be transformed into a "simple" ciphertext of a similar message. It is possible to implement the Refresh method using (a) either the bootstrapping, which accepts a C1 and C2 under a key and produces another ciphertext under another key with the same message.

- ValidityCheck(Params, EK, [C], COMP) → flag.

ValidityCheck is a deterministic algorithm. If ValidityCheck returns a flag of 1, next, the homomorphic computation COMP will yield a ciphertext that decrypts to the right answer when applied to the vector [*C*].

3.3. HE Functionality

The new algorithms will now be described in detail in the study.

Starting with the DKG algorithm, an interactive protocol between parties $\boldsymbol{p_1}, \dots, \boldsymbol{p_t}$. t will be used to implement the algorithm. A randomised algorithm, the DKG algorithm is just that. The number of parties t, as well as threshold *d* and security parameter, are inputs to DKG.
As a result of the DKG algorithm, there is a secret key vector $\boldsymbol{s} = (\boldsymbol{s_1}, \dots, \boldsymbol{s_t})$ of dimensions *t* and a public EK, whereas *t* gets (Ek,$\boldsymbol{s_i}$) from each of the other parties $\boldsymbol{p_i}$.
According to the study, p_i does not get s_j for $\boldsymbol{i \neq j}$ and $\boldsymbol{p_i}$ must keep its secret key $\boldsymbol{s_i}$ Confidential. It is followed by an

explanation of the distributed encryption (DE) algorithm. An algorithm called DE, which any party can use, p_i is a random algorithm. The plaintext M and the secret key $\boldsymbol{s_i}$ are fed into DE by party p_i. C is the ciphertext generated by DE.

A distributed-decryption (DD) procedure for an interactive protocol amongst *t* parties' subset, namely $\boldsymbol{p_1}, \dots, \boldsymbol{p_t}$ is described. It is a random algorithm that uses the DD algorithm.

Subsets of secret keys $\boldsymbol{s} = (\boldsymbol{s_1}, \dots, \boldsymbol{s_t})$, a ciphertext C and threshold parameter (d) are the inputs to DD. Furthermore, each participant's p_i gives the inputs *i*. Any party can supply the ciphertext C. Plaintext *M* is the result of the DD command.

The algorithms described above must meet the following set of requirements in terms of correctness: The DD decryption algorithm result will be correct if all parties involved in the interaction follow the stipulated interactive protocol exactly.

3.3.1. Evaluation Privacy

In addition to semantic security, the ciphertext C should be able to obscure which homomorphic computations were used to generate C. Evaluation Privacy is the term used in the study.

Cloud computing services can offer services like this, in which an ML algorithm is trained on the client's private data. Encrypted data from the client is sent to the cloud along with an evaluation key (EK); C' an encrypted version of *F*(*M*), is now being generated by the cloud and sent to a client. Evaluation privacy ensures that nothing about algorithm F can be derived from the pair (*M*, *F*(*M*)) that *C* does not reveal. Adversaries who intercept encrypted network communication may be interested in this.

3.3.2 Key Evolution

Suppose a server holds a collection of ciphertexts encrypted using a secret SK key, and the users with SK suspect if the SK is compromised. Encryption schemes should have the following key evolution feature: privacy.

The key evolution property is satisfied by any sufficiently homomorphic encryption method. Assume TK is SK encryption. Decoding TK using the secret key SK provides the plaintext SK. A server can turn ciphertext C in corpus into ciphertext C' given TK and EK by homomorphically evaluating the decryption procedure. The security of the original homomorphic encryption system arises from its semantic security. Proxy re-encryption.

3.4. Support Vector Machine

It is a supervised linear classifier known as support vector machines (SVMs). SVM could be developed to conduct nonlinear classification and classification on an unlimited number of classes with additional techniques. Using a support vector machine, you can create an arbitrary hyperplane between two sets of vectors that can be separated linearly.

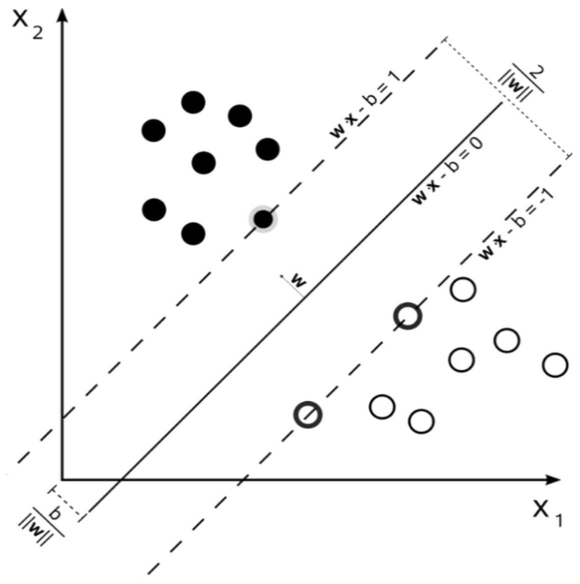

Fig. 2 Support Vector Machine

Only the hyperplane that maximizes the lowest distance from the hyperplane to any point in each cluster is considered for this task. The side of the computed hyperplane on which a testing point lies is used to classify it. Support vector machines can be extended to function in a wider range of situations, but this simple solution only works for two linearly distinct classes.

3.4.1. Non-Linearly Separable Classes

The preceding basic method does not work if the two classes cannot be separated linearly. Building a soft margin in this scenario is necessary, effectively dividing the data into two classes. A cost function is used to do this. The hinge loss function is a critical part of this cost function. If a point is on the correct side of the hyperplane, this function has no value; otherwise, it has a value proportionate to the distance the point is from the correct side of the hyperplane. The cost function combines all the data points' hinge loss functions and a parameter that defines which points are most likely to fall on either side of the hyperplane. Classification hyperplanes are generated by reducing this cost function. When the input is linearly separable, this approach is quite similar to the fundamental support vector machine algorithm, but it also works well when the input is not linearly separable.

3.4.2. Non-Linearly Separable Classes

The study looks at SVM as an example and sees how the algorithm's parameters are trained, and the data points are classified on ciphertexts. SVM is a widely used classification algorithm that allows users to create regression formulas using training data. A specific regression coefficient $z = x_1w_1 + x_2w_2 + x_nw_n + \ldots + x_nw_n$ is multiplied by each feature of the data points in the training procedure, and z is then substituted into the sigmoid function for the sigmoid function. There is no label when Sigmoid(z) is less than or equal to 0.5. Hopefully, this approach can be done in a ciphertext environment to protect critical training data. An analogous approach to training parameters on the ciphertexts based on the HE.

For this exercise, assume that w is a weight vector, that x is an input feature vector, and that the intercept is b. The encryption scheme allows the user to send ciphertexts (cw, cx, cb) to the cloud server. The server returns possible labels in ciphertext space by classifying them in the SVM. In homomorphic multiplication and addition operations, the study see that in a ciphertext, c_z is a matrix of size 6x6. Then, the sign function (c_z) is replaced with the determinant function in the case of classification (c_z). If det(c_z) is greater than zero, the label is 1; otherwise, det(c_z) equals zero.

4. Results and Discussions

In this section, we validate the model by enabling the system to detect active and side-channel attacks. The proposed model is tested on a python IDE simulator on a high-end computing system with 16GB of RAM under an intel 15-core processor. The model is validated against various metrics, including computational cost, communication cost, encryption time, decryption, and latency.

4.1. Active Attacks

Beyond the concept of semantic security, more stringent security standards can be considered. As an illustration, consider the case when a customer has data M and wants to calculate the factor $F(M)$ for a specific algorithm F. Following this procedure, a cloud computing service performs the factor F(M) computation on behalf of the client. Cloud servers get encrypted ciphertext from clients.

As an alternative, imagine that the cloud instead computes a C that was encryption of $G(M)$ for a function G. Because of the possibility of major inaccuracies, this could be a concern for the client. In this case, semantic security cannot rule out the active attack possibility on the system. According to the findings, these attacks on homomorphic encryption systems cannot be stopped without extra steps. Alternatively, the adversary may be able to decrypt the select C of its choice to gain access to the client's private information. In this case, the guarantee of security does not protect against this attack.

4.2. Side-Channel Attacks

An adversary may be able to access a portion of an encryption scheme secret key using side-channel attacks, such as timing assaults during the decryption process. Leakage resiliency is a term used to describe the ability of an encryption method to withstand such attacks. Regarding side-channel assaults, it should be difficult to breach semantic security. Leakage resistance can only withstand a limited amount of secret key leakage.

4.3. Evaluation

Fig. 3 portrays the results of computational cost between the presented HE-ML model and the existing SVM and CRF. The simulation outcomes display that the presented technique acquires a reduced rate of computation cost when compared with other existing approaches. The utilisation of ML computation enables the proposed method to achieve a reduced rate of computations.

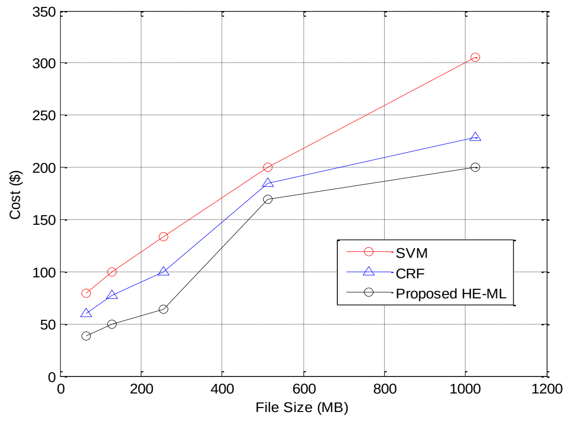

Fig. 3 Computational Cost

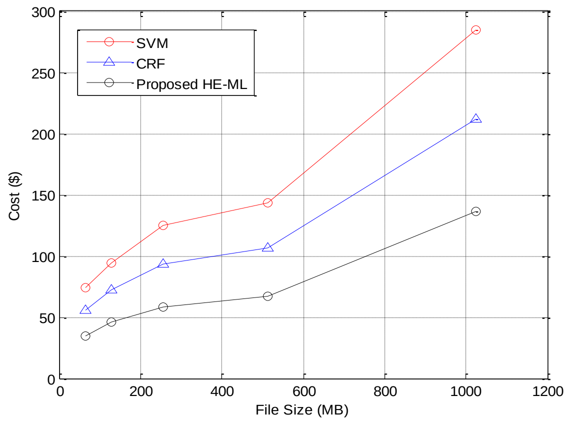

Fig. 4 Communication Cost

Fig. 4 exhibits the outcomes of communication cost between the presented HE-ML model and the existing SVM and CRF. The simulation outcomes reveal that the presented HE-ML attains a reduced rate of communication cost than other existing techniques. The utilisation of ML computation enables the proposed method to achieve a reduced communication rate.

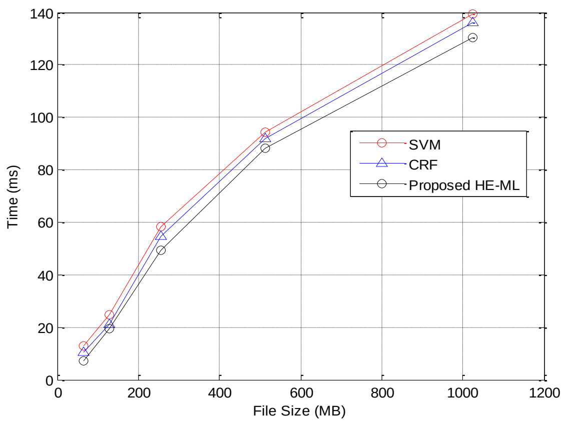

Fig. 6 Decryption time

Fig. 6 shows the results of decryption time between the proposed HE-ML model and existing SVM and CRF. The outcomes of the simulation portray the presented approach achieves a reduced rate of decryption time compared to other existing approaches. The utilisation of ML computation enables the proposed method to achieve a reduced rate of decryption time.

5. Conclusion

In this paper, ML is amended by training it with the feature vectors from various types of attacks. The model developed after training is deployed for predicting the type of incoming attack when the covid-19 data is offloaded to the cloud. Each packet is encrypted and sent to the cloud, and the possibility of an attack on these encrypted data is predicted well by the ML model. The simulations are conducted on two different datasets, including text-based and image-based, to validate the efficacy of the privacy preservation ML model in providing data privacy while the data has been classified. The simulation results show that the proposed ML encryption model enables optimal privacy of user data than other existing methods.

References

[1] Kanwal, T., Anjum, A., & Khan, A. (2021). Privacy preservation in e-health cloud: taxonomy, privacy requirements, feasibility analysis, and opportunities. Cluster Computing, 24(1), 293-317.

[2] Wu, Y., Dai, H. N., Wang, H., & Choo, K. K. R. (2021). Blockchain-based privacy preservation for 5g-enabled drone communications. IEEE Network, 35(1), 50-56.

[3] Kumar, R., Kumar, P., Tripathi, R., Gupta, G. P., & Kumar, N. (2021). P2SF-IoV: A Privacy-Preservation-Based Secured Framework for Internet of Vehicles. IEEE Transactions on Intelligent Transportation Systems.

[4] Hu, C., Cheng, X., Tian, Z., Yu, J., & Lv, W. (2021). Achieving privacy preservation and billing via delayed information release. IEEE/ACM Transactions on Networking.

[5] Parvathaneni Rajendra Kumar, et al. "Heart Disease Prediction based on Ensemble Classification Model with Tuned Training Weights." International Journal of Engineering Trends and Technology, vol. 70, no. 4, Apr. 2022, pp. 59-81.

[6] Chen, Z., Zhu, T., Xiong, P., Wang, C., & Ren, W. (2021). Privacy preservation for image data: A GAN-based method. International Journal of Intelligent Systems, 36(4), 1668-1685.

[7] Chen, F., Chen, X., Xiang, L., & Ren, W. (2021). Distributed economic dispatch via a predictive scheme: heterogeneous delays and privacy preservation. Automatica, 123, 109356.

[8] Nitesh Gaikwad, Dr. Shiyamala. S. Design and Development of Microarchitecture for Dynamic IoT Communication International Journal of Engineering Trends and Technology, 69(11), 1-8.

[9] Bansal, A., Chen, T., & Zhong, S. (2011). Privacy preserving back-propagation neural network learning over arbitrarily partitioned data. Neural Computing and Applications, 20(1), 143-150.

[10] Graepel, T., Lauter, K., & Naehrig, M. (2012, November). ML confidential: Machine learning on encrypted data. In International Conference on Information Security and Cryptology (pp. 1-21). Springer, Berlin, Heidelberg.

[11] Rahulamathavan, Y., Phan, R. C. W., Veluru, S., Cumanan, K., & Rajarajan, M. (2013). Privacy-preserving multi-class support vector machine for outsourcing the data classification in the cloud. IEEE Transactions on Dependable and Secure Computing, 11(5), 467-479.

[12] Raphael, B. O. S. T., Raluca, P., & Stephen, T. (2015). Machine learning classification over encrypted data. In Network and Distributed System Security Symposium. NDSS Symposium, San Diego.

[13] Wong, W. K., Cheung, D. W. L., Kao, B., & Mamoulis, N. (2009, June). Secure knn computation on encrypted databases. In Proceedings of the 2009 ACM SIGMOD International Conference on Management of data (pp. 139-152).

[14] Aslett, L. J., Esperança, P. M., & Holmes, C. C. (2015). Encrypted statistical machine learning: new privacy preserving methods. arXiv preprint arXiv:1508.06845.

[15] Liu, X., Lu, R., Ma, J., Chen, L., & Qin, B. (2015). Privacy-preserving patient-centric clinical decision support system on naive Bayesian classification. IEEE journal of biomedical and health informatics, 20(2), 655-668.

[16] Gilad-Bachrach, R., Dowlin, N., Laine, K., Lauter, K., Naehrig, M., & Wernsing, J. (2016, June). Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In International conference on machine learning (pp. 201-210). PMLR.

[17] Y.Krishna Bhargavi, Dr. Yelisetty Ssr Murthy, Dr.O.Srinivasa Rao "A Real And Accurate Semantic Search Indexing Approach Using Asvm Machine In Big Data Analytics" International Journal of Engineering Trends and Technology 69.2(2021):144-159.

[18] Li, X., Zhu, Y., Wang, J., Liu, Z., Liu, Y., & Zhang, M. (2017). On the soundness and security of privacy-preserving SVM for outsourcing data classification. IEEE Transactions on Dependable and Secure Computing, 15(5), 906-912.

[19] Aono, Y., Hayashi, T., Wang, L., & Moriai, S. (2017). Privacy-preserving deep learning via additively homomorphic encryption. IEEE Transactions on Information Forensics and Security, 13(5), 1333-1345.

[20] Kim, M., Song, Y., Wang, S., Xia, Y., & Jiang, X. (2018). Secure logistic regression based on homomorphic encryption: Design and evaluation. JMIR medical informatics, 6(2), e8805.

[21] Liu, X., Choo, K. K. R., Deng, R. H., Lu, R., & Weng, J. (2016). Efficient and privacy-preserving outsourced calculation of rational numbers. IEEE Transactions on Dependable and Secure Computing, 15(1), 27-39.

[22] Sun, X., Zhang, P., Liu, J. K., Yu, J., & Xie, W. (2018). Private machine learning classification based on fully homomorphic encryption. IEEE Transactions on Emerging Topics in Computing, 8(2), 352-364.

[23] Maheshwari, V., Mahmood, M. R., Sravanthi, S., Arivazhagan, N., ParimalaGandhi, A., Srihari, K., ... & Sundramurthy, V. P. (2021). Nanotechnology-Based Sensitive Biosensors for COVID-19 Prediction Using Fuzzy Logic Control. Journal of Nanomaterials, 2021.

[24] Khadidos, A., Khadidos, A. O., Kannan, S., Natarajan, Y., Mohanty, S. N., & Tsaramirsis, G. (2020). Analysis of COVID-19 Infections on a CT Image Using DeepSense Model. Frontiers in Public Health, 8.

[25] Raja, R. A., Karthikeyan, T., & Kousik, N. V. (2020). Improved Privacy Preservation Framework for Cloud-Based Internet of Things. In Internet of Things (pp. 165-174). CRC Press.

[26] Baryalai, M., Jang-Jaccard, J., & Liu, D. (2016, December). Towards privacy-preserving classification in neural networks. In 2016 14th annual conference on privacy, security and trust (PST) (pp. 392-399). IEEE.