2023 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom) | 979-8-3503-3782-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/BLACKSEACOM58138.2023.10299735

Privacy-Preserving User Abnormal Behavior Detection in 5G Networks

Berker Acir*, Ertan Onur*,†, Raouf Boutaba†
*Department of Computer Engineering, Middle East Technical University
{berker.acir | eronur}@metu.edu.tr
†David R. Cheriton School of Computer Science, University of Waterloo
rboutaba@uwaterloo.ca

***Abstract*—Network Data Analytics Function (NWDAF) plays a key role in autonomous network management and security using machine learning (ML) techniques. However, not all ML methods such as convolutional neural networks, deep language models can be deployed in the core network due to their high computational requirements. Besides, telcos may prefer handing over ML analytics to experienced third parties instead of implementing themselves. In this paper, we show that delay-tolerant tasks of NWDAF can be offloaded to public third party cloud providers while preserving user and data privacy by employing homomorphic encryption (HE). We focus on the problem of abnormal behaviour detection for a group of user equipment (UE) and train various DL models with plaintext data. We demonstrate that inferencing on encrypted data is as accurate and precise as plaintext inferencing and we quantify the performance degradation in terms of running times. We believe that offloading NWDAF tasks to the cloud allows telcos to focus on expanding their core capabilities, and reduce capital and operational expenditures.**

***Index Terms*—5G, NWDAF, Privacy-Preserving, Deep Learning, Homomorphic Encryption**

I. Introduction

The increasing number of connected devices and the emergence of new applications with diverse requirements bring unprecedented challenges to both mobile access and core networks [1]. These challenges pertaining to the need for flexibility and scalability are due to the fact that current mobile network functions are strongly coupled with specialized hardware [2]. 5G mobile networks are designed following a service-oriented architecture where network elements are decomposed and modularized into network functions that offer their services via restful web interfaces [3]–[5]. In this approach, software-defined networking (SDN) [6] and network function virtualization (NFV) [7] are respectively employed to decouple the network control and data planes and to implement network functions in virtual machines or containers running on commodity hardware. Leveraging cloud and edge computing, 5G networks achieve enhanced flexibility, scalability and deployability compared to traditional networks, and are able to meet diverse application requirements thanks to network programmability [8].

This study is supported by Turkcell Technology Research and Development within the framework of 5G and Beyond Joint Graduate Support Programme coordinated by Information and Communication Technologies Authority. Dr. Onur is supported by TÜBİTAK BİDEB 2219 grant (1059B192200291).

The above flexibility and programmability come at the cost of increased management complexity for network operators. Network automation using data-driven analytics is widely adopted as a means to tame management complexity. Towards this aim, Network Data Analytics Function (NWDAF) has been standardized in Release 15 of 3GPP and is continuously enhanced in subsequent releases. NWDAF collects network statistics, metrics and events automatically and produces insights about the network. NWDAF takes snapshots of the network dynamically; it measures the performance of services and functions, models and processes the data, and shares analytics results with other network functions. Meaningful network and user information is created from the collected data using artificial intelligence (AI), machine learning (ML) and data mining techniques [9]. This way NWDAF facilitates autonomous network operation, efficient resource management, enhanced service quality, reliability and security.

NWDAF, as defined in Release 17, consists of the following logical functions for leveraging AI/ML: Analytics Logical Function (AnLF) and Model Training Logical Function (MTLF). AnLF is responsible for performing inferences, deriving and exposing analytics. AnLF requires the models from MTLF whose responsibility is to train and deploy ML models and exposing training service. Due to dynamic nature of the network and data, MTLF self-monitors provided models' performances and initiates re-training or re-designing processes when required. In Release 17 [9], there can be more than one NWDAF instance in the network, and an instance has either AnLF, MTLF or both functionalities combined. An NWDAF instance can be the consumer of another instance. For example, NWDAF instance with only AnLF can make use of provided models from other instances with MTLF.

Most studies on NWDAF assume that network providers cannot rely on third party cloud computing for their data analytics services due to data privacy concerns. However, recent advances in cryptography have made it possible to both train ML models and make inferences without violating data privacy. Offloading NWDAF functionalities, especially delay-tolerant ones, to third party cloud providers will enable network operators to be more agile, concentrate on their core capabilities while reducing hardware requirements and keeping up with the ever-growing demands of resources [10]. This is thanks to with cloud elasticity and ability to provision

resources on demand.

There may be cases where ML models, especially Deep Learning (DL) models, cannot be deployed in the core network due to their high computational needs. Additionally, network operators may prefer to delegate implementation of ML models to third parties that have expertise in this area rather than implementing them by themselves. In such cases, NWDAF could rely on models that are trained or deployed in public clouds. A significant issue that can arise from using public cloud computing services is data privacy since cloud services may be vulnerable to data breaches [11] and insider attacks [12]. Preserving data privacy can be achieved with homomorphic encryption (HE) that allows to perform computations on encrypted data thereby eliminating the risk of exposing sensitive data.

As the main contribution of this paper, we show that NWDAF can employ DL models on public cloud servers while preserving privacy of data and their owners by employing HE. To demonstrate the feasibility of this approach, we focus on the problem of detecting abnormal user behaviour. We trained various DL models for abnormal user behaviour detection and evaluate their performance and limitations when using DL with HE. Furthermore, we quantify the performance degradation traded off for preserving user and data privacy.

This paper is organized as follows. Section II presents the user abnormal behaviour detection problem, discusses related work, our proposed architecture for privacy-preserving analytics on NWDAF, and explores investigate HE libraries and frameworks compatible with DL while describing the dataset and our DL models. Section III introduces our evaluation methodology, describes performed experiments and discussed obtained results. Finally, Section IV concludes the paper and points out future directions for privacy-preserving NWDAF analytics.

II. Anomaly Detection for a Group of User Equipment (UE)

The focus of this paper is on one of the use cases of NWDAF, namely the Abnormal Behaviour Information (anomaly) for a Group of UEs which is believed to be crucial for network security and service quality. Hijacked or misused UEs can be the cause of abnormal behaviour, e.g. abnormal traffic patterns, unexpected UE locations, unexpected transaction dispersion amounts, wrong destination addresses. 3GPP does not define how NWDAF will detect those abnormalities and leaves the implementation to service providers.

In [13], Yuan et al. provide an extensive literature review of anomaly detection techniques in 5G with or without considering NWDAF. The reviewed anomaly detection techniques include supervised, semi-supervised and unsupervised ML techniques. Each has its own advantages and disadvantages. Supervised learning methods such as DL can be effective in achieving more accurate detection but require labeled data. Semi-supervised and unsupervised learning methods are particularly useful when there are insufficient or no labeled data. However, their performances can be insufficient for example when the data have high intra-class differences.

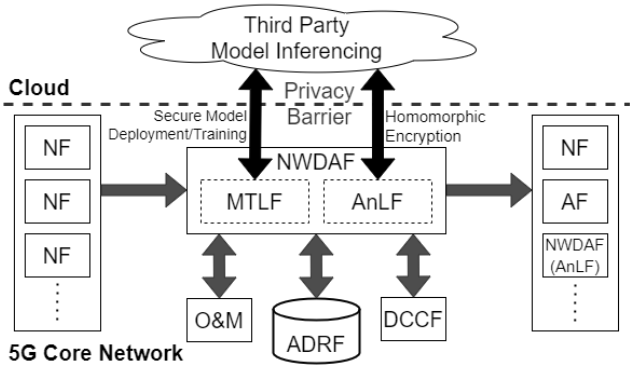


Figure 1: NWDAF architecture where MTLF either securely deploys trained model to the cloud or initiates secure model training on the cloud, and AnLF makes inferences on the cloud models with encrypted inputs.

Although ML heavily depends on data, the literature still lacks comprehensive datasets for NWDAF use cases even though network data analytics services were first introduced in 3GPP Rel-15 late 2017. In [14], Sevgican et al. address this issue by providing a publicly available synthetic dataset[1] which can be used for two of the standardized NWDAF use cases: network load level prediction; and anomaly detection. We discuss details of the dataset in Section II-B and how it is in this work. The authors also provide a supervised ML solution for network load prediction and network anomaly classification. Linear regression, long short-term memory (LSTM), and recursive neural network (RNN) are used for prediction whereas logistic regression and eXtreme Gradient Boosting (XGBoost) are used for anomaly detection [14].

In our work, we devise the architecture shown in Figure 1 for Abnormal Behaviour Detection for a Group of UEs where NWDAF instances make use of third party cloud computing services while preserving data-privacy. MTLF's secure model deployment to cloud services or its secure model training initiation process are not detailed in this paper for the sake of conciseness. Model training is done on plaintext data and we focused on AnLF's secure model inferences conducted on encrypted data using homomorphic encryption. We evaluate the performance of this approach under various configurations and compare our results with the work of Sevgican et al [14].

A. Privacy Preservation in Deep Learning

The theory of HE dates back to 1970s [15] but was considered impractical. However, the breakthrough by Craig Gentry [16] in 2009 and subsequent improvements in both theory and implementations together with general hardware advances paved the way for the use of HE in practice with performance improvements by up to five orders of magnitude [17]. For instance, multiplication time of ciphertexts has been decreased from 30 minutes to less than 20 milliseconds.

[1] https://github.com/sevgicansalih/nwdaf_data

Advances in HE enabled many real-world applications in various domains but most importantly it led to the development of HE libraries, compilers and frameworks empowering non-expert in cryptography to develop their own secure and privacy-preserving applications. The authors of [17] describe these HE libraries, compilers and frameworks and how they are used in their evaluations[2].

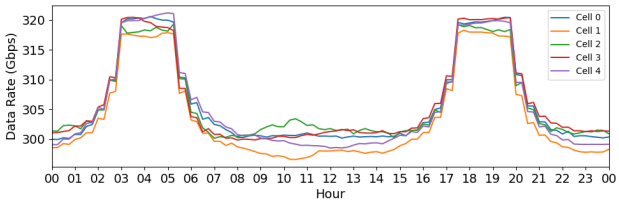
In our work, we adopt the Intel's graph compiler for Artificial Neural Networks with HE-backend (nGraph-HE) [18] [19] which supports CKKS encryption scheme implemented by the Simple Encrypted Arithmetic Library (SEAL) from Microsoft Research [20]. This choice is motivated by the fact that it simplifies working on pre-trained neural networks for making privacy-preserving inferences by abstracting FHE operations such as key generation, encryption, and decryption.

Even though the framework simplifies private inferencing, there are engineering challenges related to the framework, SEAL library and HE. One of the challenges comes from HE where HE computations are, by definition, data independent and this restricts branching on data. This also prevents the usage of significant DL methods such as the ReLU activation function, or the minimum and maximum pooling layers in convolutional neural networks. Such DL models trained under these constraints are referred to as HE-friendly models. Another challenge comes from configuring SEAL which offers leveled HE operations. A wrong configuration of the parameters will result in private inferencing different from what is expected since the library does not implement bootstrapping which is an important technique for refreshing ciphertexts to reduce the noise to a fixed level.

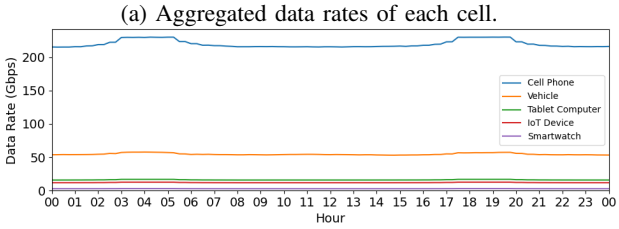
B. Dataset

The synthetic dataset used has 6 fields: time period, network area information as cell identifier, subscription category, user equipment type, data rate and anomaly information. A row of the data contains the information about data rate of group of personal equipment within the specific subscription category connected to a particular cell in a specific time period together with whether there is an anomaly or not. There are five types of personal equipment: IoT device, vehicle, cell phone, smart watch, and tablet computer. There are three subscription categories which are platinum, gold, and silver. The authors of the dataset used fixed topology with 5 cells together with their adjacency information, and they applied mobility of user equipment when generating the data while considering personal equipment type, subscription category and time of day. For example, mean handover ratio of IoT device is much more lower than that of the vehicle due to the nature of these equipment, and mean handover ratios are the lowest between 22:00 and 06:00 as humans are less likely to move around at night time. Sevgican et al. considered rush hours (07:00-09:30 and 16:00-20:00) where personal equipment have lower handover ratios compared to other time of the day except night time since the increased traffic

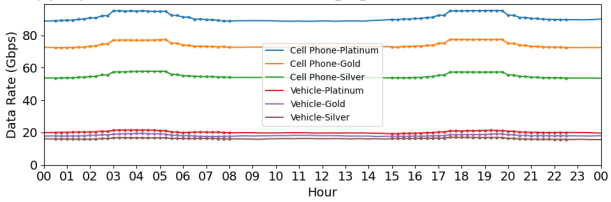
[2]https://github.com/MarbleHE/SoK



(a) Aggregated data rates of each cell.



(b) Aggregated data rates of each equipment connected to cell 1.



(c) Data rates of cell phones and vehicles by their subscription categories together with anomalies.

Figure 2: One day sample from the dataset redrawn from [14].

affects our daily lives and mobility. In addition, data rates vary between subscription categories and personal equipment. For example, people tend to buy better subscription categories for their cell phones rather than for their IoT devices.

Cell phones have the highest data rate while smart watches have the lowest data rate. The authors generate the dataset by aggregating each type of personal equipment with their subscription category within the cell. Then, they simulate the mobility of each personal equipment with their subscription categories with specified handover ratios between adjacent cells, and their aggregated data rates are used in dataset. In addition, they add anomalies to specific time periods which results in unexpected increase in network traffic and the anomaly fades and stabilizes in time.

Aggregated data rates of each cells can be seen in Figure 2a where the spikes represent anomalies in the data. In Figure 2b, we show the data rates of each personal equipment type connected to cell 1. Data rate of cell phone is the highest among personal equipment types. In Figure 2c, we show the data rates of cell phone and vehicle connected to cell 1 and their data rates are separated by their subscription categories. Dots on the curves show that there is an anomaly in the data.

C. Deep Learning Models

Dataset has been pre-processed before used in the DL models. Time and load values are normalized between zero and one. Also, RRU cell, subscription category and personal equipment identifiers are converted into one-hot vectors. The main reason for pre-processing of dataset is to decrease

overfitting and underfitting of HE-friendly neural networks due to their limited capabilities. We observed that normalizing the data and using one-hot vectors for categorical data had improved the trained models accuracy, and also, decreased overfitting, and especially, underfitting probabilities. Since HE-friendly neural networks cannot directly use one of the most essential part of the deep learning models, namely the ReLU activation function, the data was underfitting to our models. Even though, we have not used convolutional layers in our model, HE-friendly DL models could not directly make use of some of the pooling types such as minimum and maximum pooling since branching is not possible on the encrypted data when HE is employed.

Considering the restrictions of HE and DL, our models use dense layers followed with polynomial activation function except the last layer. Outputs of the last layer is used in categorical cross-entropy loss function with the Adam optimizer with default values (0.001 learning rate, 0.9 $\beta 1$ and 0.999 $\beta 2$) as a stochastic gradient descent (SGD) method. Our models take input data such as time, load, RRU cell, subscription category and personal equipment information to predict whether the given data has anomaly or not. Polynomial activation functions have two learnable parameters as follows. The weights (i.e., learnable parameters) have been initialized with the He normalization, also known as Kaiming initialization introduced by He et al. in [21]. Both L1 and L2 regularization methods are applied on kernel and bias weights in order to reduce overfitting and underfitting.

Another limitation of the HE is that the number of multiplication operations performed on the encrypted data is limited as the error (noise) on ciphertext accumulates with each multiplication and the result will eventually become unreliable and impossible to decrypt. Hence, the number of dense and activation layers are kept low.

After the plaintext model training and validation, nGraph-HE framework is used for plaintext and ciphertext inferencing. First, the framework converts Tensorflow computations into computational graph and depending on the experiment type (i.e., plaintext or ciphertext inferencing), the framework converts computations on the graph into related operations using SEAL. For example, multiplications between inputs and weights are converted into HE multiplication operation between ciphertext and plaintext implemented by SEAL. Then, the framework configures key generation, encryption and decryption processes of input if the framework is configured to work with HE scheme.

III. Results and Discussion

The ultimate goal of this paper is to show that encrypted data can be used for inference and achieve similar results to previous works that use plaintext data. We argue that some of the functionalities of NWDAF in 5G networks can be offloaded to third party cloud providers while preserving data privacy. In this perspective, we used nGraph-HE framework to test trained DL models for inferencing on encrypted data. In addition to comparing ciphertext inferencing with plaintext inferencing in terms of success and computational time overhead, experiments with various configurations are conducted to analyze impact on the results and runtimes.

A. Methodology

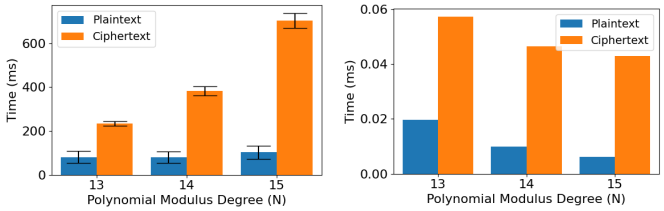
We used 70% of the dataset for training, 21% for validation, and 9% for evaluating the performance of our models on both plaintext and ciphertext data. Multiple neural network models with similar architectures, but with a different number of layers and parameters are evaluated together with a combination of encryption parameters. There are a total of 102 different experiment configurations where each configuration was run at least 20 times, and the results such as running times, accuracy and precision were averaged.

nGraph-HE and SEAL related encryption parameters are the degree of the polynomial modulus, coefficient modulus, security level and complex packing. The polynomial modulus degree represents the degree of a power-of-two cyclotomic polynomial. Coefficient modulus is an integer modulus which is constructed as a product of multiple distinct prime numbers whose bit-lengths are described in the configuration. Security level can be 128, 192 and 256 bits representing the security level of the encryption scheme. With the complex packing enabled, throughput is doubled by taking advantage of the complex components of the plaintext encoding map. The choice of encryption parameters significantly affects the performance, capabilities, and security of the encryption scheme. Using a larger polynomial modulus together with complex packing will allow to work on more elements without decreasing the security level while decreasing the performance.

Experiments were run on an Intel Xeon Silver 4114 server with 2.20GHz CPU and 64GB RAM. In the experiments, different models are trained and the resulting models are used with different configurations in plaintext and ciphertext inferencing. There are 6 different models where there are 2 different layer counts together with 3 different parameter count options. Using models with 5 and 7 layers were observed to be sufficient for this task. The former model has 3 dense layers followed with 2 activation layers except the last (output) layer whereas the latter model has 4 dense layers followed with 3 activation layers except the last layer. The number of the layers can be increased when dealing with much more complex problems, but polynomial modulus and coefficient modulus parameters must be configured accordingly since arithmetic operations on the encrypted data accumulates the error, and the results may deviate after a certain number of operations due to lack of bootstrapping. In the sequel, we present the experimental results to analyze the impact of various parameters of HE on the runtime.

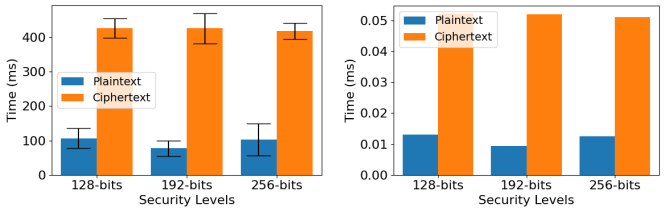
B. Impact of Polynomial Modulus Degree, N

In Figure 3, we present the averaged and amortized runtimes of a model with 5 layers and 452 trainable parameters and the ciphertext inferencing is done with 128-bits security level and three different polynomial modulus degrees (N): 13, 14, and 15. Since polynomial modulus degree determines



(a) Averaged runtimes. (b) Amortized runtimes.

Figure 3: Running times of a DL model with 5 layers and 452 parameters with 128-bits security where three different polynomial modulus (N) degrees are compared.



(a) Averaged runtimes. (b) Amortized runtimes.

Figure 4: Running times of a DL model with 5 layers and 888 parameters with polynomial modulus degree of 14 where three different security levels are compared.

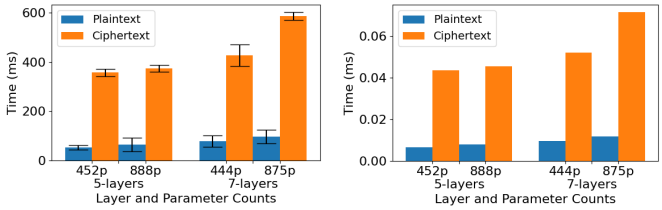
the maximum input size (2^{N-1}), it is observed that increasing polynomial modulus degree increases the total runtime. However, amortized runtime of an input decreases with the increase of polynomial modulus degree.

C. Impact of the Security Level

In Figure 4, we show the averaged and amortized runtimes of a model with 5 layers and 452 trainable parameters and the ciphertext inferencing is done with polynomial modulus degree of 14 and three different security levels: 128-bits, 192-bits and 256-bits. It is observed that changing security level does not affect the total runtime. It is important to keep the same number of the coefficient modulus. Otherwise, the configuration with higher number of coefficient modulus will perform slower regardless of its security level.

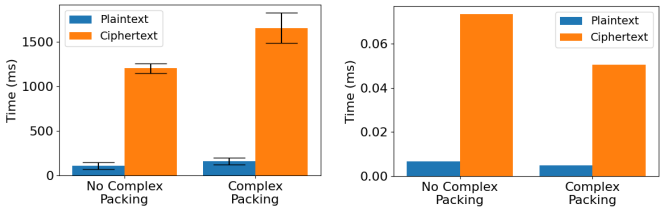
D. Impact of DL Layers

In Figure 5, the averaged and amortized runtimes of four different models with 5 and 7 layers where their trainable parameter counts are close to 450 and 880 are presented, and the ciphertext inferencing is done with polynomial modulus degree of 15 and security level of 192-bits. It is observed that introducing new layers into the DL models or increasing the parameter count in the DL models increase the running times. The parameter count increases the number of arithmetic operations (i.e., multiplication, summation) executed on the encrypted data. However, the layer count has much more impact on the running times as it also increases the multiplicative depth of the encrypted data.



(a) Averaged runtimes. (b) Amortized runtimes.

Figure 5: Running times of 4 different DL models with layer counts of 5 and 7, approximately 450 and 880 parameters with polynomial modulus degree of 15 and 192-bits security level where numbers of parameters and layers are compared.



(a) Averaged runtimes. (b) Amortized runtimes.

Figure 6: Running times of a DL model with 7 layers and 1200 parameters with polynomial modulus degree of 15 and 256-bits security level where complex packing feature is compared.

E. Impact of Complex Packing

In Figure 6, we present the averaged and amortized runtimes of testing the complex packing option with a model of 7 layers and 1200 parameters where ciphertext inferencing is done with polynomial modulus degree of 15 and 256-bits security level. When complex packing is disabled, the maximum input number is equal to 2^{N-1} where N is the polynomial modulus degree. The maximum input number is doubled when complex packing is enabled. Therefore, averaged running time increases with enabling complex packing. Yet, complex packing reduces the amortized running time.

F. Evaluation of the Privacy Preservation

The average time for training a DL model with plaintext data together with validation step is 24.98 ± 2.87 seconds where it takes 5 epochs to train a DL model. As shown in the experiment results, inferencing on encrypted data with HE increases computational time cost due to the nature of HE. Averaged and amortized runtimes of inferencing on plaintext and ciphertext data are shown in the Table I. Privacy preserving inferencing with HE increases computational runtimes by the magnitude of 5 in terms of amortized runtimes whereas it increases running time by the magnitude of 6 on the average without considering amortized running times.

Average accuracy and precision values for training, validation and testings of all experiments are shown in Table II. Average plaintext and ciphertext testing accuracy and precision values are equal to 0.776 ± 0.01. Average approx-

imation value between plaintext and ciphertext inferencing is $1.271 \times 10^{-8} \pm 2.59 \times 10^{-8}$. It can be concluded that the accuracy and precision difference between plaintext and ciphertext output values is negligible.

Table I: Averaged and amortized time results of experiments.

	Plaintext	Ciphertext
Time	103.47 ± 44.57 ms	653.47 ± 376.13 ms
Amortized	0.01 ± 0.006 ms	0.049 ± 0.018 ms

Table II: Experiments' average accuracy and precision results.

	Training	Validation	Plaint. & Ciphert.
Accuracy	0.775 ± 0.002	0.775 ± 0.006	0.776 ± 0.01
Precision	0.775 ± 0.004	0.774 ± 0.006	0.776 ± 0.01

Sevgican et al. states in [14] that their base logistic regression algorithm achieves 0.556 accuracy and 0.769 precision while their XGBoost algorithm achieves 0.626 accuracy and 0.773 precision. Average plaintext and ciphertext testing accuracy and precision values are 0.776 ± 0.01. Comparing the results of both works, it can be concluded that using encrypted data with HE for inferencing does not decrease the trained model's accuracy and precision although it introduces a running time overhead. The abnormal behaviour detection for a group of UEs can be executed in a public cloud while preserving data privacy. However, the use of HE is restricted to the analysis of historic data and we have to enhance the performance of HE-based schemes for real-time application settings.

IV. Conclusions

This paper proposed offloading delay-tolerant tasks of NWDAF to third party (public) cloud service providers while preserving data privacy using homomorphic encryption schemes. This allows telecom operators to be more agile, to focus on their core capabilities, and to reduce capital and operational expenditures. Even though NWDAF was first introduced in late 2017, the literature still lacks network data that can be used for NWDAF use cases. In light of this shortcoming, we have used a synthetic NWDAF dataset which contains abnormal behavior information for a selected group of UEs, and various deep learning models are trained and tested with both plaintext and ciphertext data. We showed that inferencing with ciphertext data is as successful as inferencing with normal plaintext data and its accuracy and precision are similar to or slightly better than results of previous works with the same data. In addition, we evaluated both runtime and amortized running time effects of encryption and DL parameters through extensive experiments and show that preserving privacy increases both averaged amortized runtimes and running times by 5 and 6 times, respectively.

In the future, our work can be extended to use ciphertext data for both model training and inferencing in the cloud, and to evaluate communication costs, security threats along with the computational overhead. These extensions will further affirm our belief that both model training and analytical functionalities of NWDAF instances can be offloaded to third party cloud providers while preserving the privacy of the network and user data.

References

[1] M. Shafi, A. F. Molisch, P. J. Smith, T. Haustein, P. Zhu, P. De Silva, F. Tufvesson, A. Benjebbour, and G. Wunder, "5G: A tutorial overview of standards, trials, challenges, deployment, and practice," *IEEE JSAC*, vol. 35, no. 6, pp. 1201–1221, 2017.

[2] Y. Li and M. Chen, "Software-defined network function virtualization: A survey," *IEEE Access*, vol. 3, pp. 2542–2553, 2015.

[3] I. F. Akyildiz, P. Wang, and S.-C. Lin, "Softair: A software defined networking architecture for 5G wireless systems," *Computer Networks*, vol. 85, pp. 1–18, 2015.

[4] H. Wang, S. Chen, H. Xu, M. Ai, and Y. Shi, "Softnet: A software defined decentralized mobile network architecture toward 5G," *IEEE Network*, vol. 29, no. 2, pp. 16–22, 2015.

[5] F. Mademann, "The 5G system architecture," *Journal of ICT Standardization*, pp. 77–86, 2018.

[6] T. Chen, M. Matinmikko, X. Chen, X. Zhou, and P. Ahokangas, "Software defined mobile networks: concept, survey, and research directions," *IEEE Communications Magazine*, vol. 53, no. 11, pp. 126–133, 2015.

[7] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Communications Magazine*, vol. 53, no. 2, pp. 90–97, 2015.

[8] L. Ma, X. Wen, L. Wang, Z. Lu, and R. Knopp, "An sdn/nfv based framework for management and deployment of service based 5G core network," *China Communications*, vol. 15, no. 10, pp. 86–98, 2018.

[9] 3GPP, "Architecture enhancements for 5G System (5GS) to support network data analytics services," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 23.288, May 2022, version 17.4.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3579

[10] N. Salhab, R. Rahim, R. Langar, and R. Boutaba, "Offloading network data analytics function to the cloud with minimum cost and maximum utilization," in *Proc. of the ICC 2020*, 2020, pp. 1–6.

[11] L. Cheng, F. Liu, and D. Yao, "Enterprise data breach: causes, challenges, prevention, and future directions," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 7, no. 5, p. e1211, 2017.

[12] A. J. Duncan, S. Creese, and M. Goldsmith, "Insider attacks in cloud computing," in *Proc. of the IEEE TRUSTCOM*, 2012, pp. 857–862.

[13] Y. Yuan, C. Gehrmann, J. Sternby, and L. Barriga, "Insight of anomaly detection with nwdaf in 5G," in *Proc. of the CITS*, 2022, pp. 1–6.

[14] S. Sevgican, M. Turan, K. Gökarslan, H. B. Yilmaz, and T. Tugcu, "Intelligent network data analytics function in 5G cellular networks using machine learning," *Journal of Communications and Networks*, vol. 22, no. 3, pp. 269–280, 2020.

[15] R. L. Rivest, L. Adleman, M. L. Dertouzos *et al.*, "On data banks and privacy homomorphisms," *Foundations of secure computation*, vol. 4, no. 11, pp. 169–180, 1978.

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

[17] A. Viand, P. Jattke, and A. Hithnawi, "SoK: Fully homomorphic encryption compilers," in *2021 IEEE Symposium on Security and Privacy (SP)*, 2021, pp. 1092–1108.

[18] F. Boemer, Y. Lao, R. Cammarota, and C. Wierzynski, "Ngraph-he: A graph compiler for deep learning on homomorphically encrypted data," in *Proc. of the 16th ACM CF*, ser. CF '19, NY, USA, 2019, p. 3–13.

[19] F. Boemer, A. Costache, R. Cammarota, and C. Wierzynski, "Ngraph-he2: A high-throughput framework for neural network inference on encrypted data," in *Proceedings of the 7th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, ser. WAHC'19. New York, NY, USA: Association for Computing Machinery, 2019, p. 45–56. [Online]. Available: https://doi.org/10.1145/3338469.3358944

[20] "Microsoft SEAL (release 4.1)," https://github.com/Microsoft/SEAL, Jan. 2023, microsoft Research, Redmond, WA.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. of the IEEE ICCV*, December 2015.