A Privacy-Preserved Variational-Autoencoder for DGA Identification in the Education Industry and Distance Learning

Xingxing Zheng and Xiaona Yin

Zhengzhou Preschool Education College, Zhengzhou 450000, China

Correspondence should be addressed to Xingxing Zheng; xingxing_zheng@126.com

Received 16 February 2022; Revised 23 February 2022; Accepted 26 February 2022; Published 24 March 2022

Academic Editor: Konstantinos Demertzis

Copyright © 2022 Xingxing Zheng and Xiaona Yin. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

One of the most insidious methods of bypassing security mechanisms in a modern information system is the domain generation algorithms (DGAs), which are used to disguise the identity of malware by periodically switching the domain name assigned to a command and control (C&C) server. Combating advanced techniques, such as DGAs, is an ongoing challenge that security organizations often need to work with and possibly share private data to train better and more up-to-date machine learning models. This logic raises serious concerns about data integrity, trade-related issues, and strict privacy protocols that must be adhered to. To address the concerns regarding the privacy and security of private data, we propose in this work a privacy-preserved variational-autoencoder to DGA combined with case studies from the education industry and distance learning, specifically because the recent pandemic has brought an explosive increase to remote learning. This is a system that, using the secured multi-party computation (SMPC) methodology, can successfully apply machine learning techniques, specifically the Siamese variational-autoencoder algorithm, on encrypted data and metadata. The method proposed for the first time in the literature facilitates learning specialized extraction functions of useful intermediate representations in complex deep learning architectures, producing improved training stability, high generalization performance, and remarkable categorization accuracy.

1. Introduction

The pandemic has had an impact on how people learn and on all stakeholders involved. The functioning of higher education institutions has been harmed. Many institutions are currently unable to perform examinations because of the pandemic, and the face-to-face teaching and learning process has been harmed. The current scenario is projected to last for some time, so it is critical to improve the learning process by establishing strategies with respect to online learning, preserving social distance, and, of course, the privacy of the data exchanged online by the various stakeholders [1, 2]. Researchers have been focusing on the growth of distance education in recent years, but because of the COVID-19 pandemic, distance learning has become a critical task for the education system and the privacy related issues.

For several years, botnet technology has been the mainstay of orchestrating and supporting a wide variety of cyber-attacks, such as DDoS attacks, and phishing. Malicious botnet masters use DGAs extensively to make it possible for the C&C server to communicate with their bots in such a way as to bypass the known malware detection mechanisms. Typically, a DGA algorithm, using a seed known only to devices participating in the botnet, periodically generates, at random times, a pseudorandom set of domain names, which act as candidates for assignment to the C&C server [3]. As a result, traditional static botnet handling techniques are becoming ineffective as the pace at which C&C changes names is unable to detect and terminate communication in time. Examples of such static techniques include blacklisting the static domain name of a C&C server as soon as it is detected, reverse-engineering the malware of an infected device to reconstruct the DGA used roughly and
studying how the malware generates names. Ideally, however, malicious names should be detected in real-time, with the predictions being performed at the level of individual names, to avoid establishing bots’ communication with the C&C server [4].

The field of machine learning has dramatically attracted the interest of cyber security researchers to address this problem [5]. In previous approaches based on machine learning methods, a DGA [6] name detector required the extraction of human-defined statistical attributes to be effective. Instead, machine learning techniques automatically extract the necessary features during the training process while relying solely on the domain name string to make the required predictions, categorizing the names between DGA and legit. This feature is handy as malware is no longer aware of the components used to train model detectors. As a result, they cannot modify DGAs to generate names that are not detected based on these characteristics. Models trained in machine learning techniques are highly accurate and efficiently generalized, as presented in the relevant literature [7, 8].

The rest of the work includes Section 2, which provides an overview of the methods found in the literature and related to similar methods. Section 3 describes in detail the motivation of this work. Section 4 includes the methodology of the proposed system. Section 4 explains the dataset used, and Section 5 analyzes the experiments for implementing the proposed approach. Finally, Section 6 summarizes the research conducted and presents the future objectives that can extend it.

2. Literature Review

With the increasing use of modern technologies in every field globally, the need to enhance the cyber security posture of educational organizations has significantly grown in recent years, especially in 2020 and after the COVID-19 pandemic. Many modern institutions have shifted their services to a remote-based approach. The research community has also focused on the education industry and distance learning to find innovative and privacy-preserving solutions to this new reality [2, 9].

Zhang et al. [10] reviewed current research on privacy-preserving technologies and deep collaborative learning. They concluded that each privacy-preserving technology has its own unique features. They asserted that while safe multiparty computing and homomorphic encryption can provide a high level of privacy and accuracy, the cost to users is a substantial computational and communication burden. Differential privacy, in which users input random noise into their data before sending it to the server, is a more practical and efficient solution. Nonetheless, it diminishes the model’s accuracy. When huge businesses with sensitive data operate as users, homomorphic encryption technology is essential to assure the model’s security. Differential privacy technology is required when a large number of people with little computer power operate as consumers to ensure the model’s efficiency. A mix of safe multiparty computing, homomorphic encryption, and differential privacy is being used in a growing number of studies to provide a suitable trade-off between data privacy and utility.

Ryffel et al. [11] developed and explored a federated learning framework built on PyTorch for privacy-preserving deep learning. Their framework prioritized data ownership and secure processing. It proposed a representation based on command chains and tensors, allowing them to perform complicated privacy-preserving structures like federated learning, SMPC, and differential privacy while presenting a familiar deep learning API to the end-user. The Boston Housing and Pima Indian Diabetes datasets were used to test their implementation. Nonetheless, they discovered early in the development process that the framework added an enormous performance burden.

Bo Chang [12] used vignettes to depict various privacy scenarios in the online learning environment while researching student privacy issues in online learning environments. Because of the sharing of individual grades among group members and providing public input on blogs, his studies revealed direct legal ramifications of concern to students that are not entirely addressed in FERPA policy. Other privacy issues, such as open access to each other’s work, transparent reflections, public comments, critical examination of assignments, and collaborative evaluations of students’ work, arose in more nuanced ways. He suggested focusing on students’ products rather than their names, informing students of the benefits they will receive and the contributions they will make by sharing their work publicly, and providing options for students to keep their identities private if they are uncomfortable about sharing their work publicly. To alleviate students’ discomfort, he also emphasized that professors should educate students about the FERPA policy while adapting to partial privacy.

In his research study, Karunakaran [3] employed both public and real-time environmental datasets to detect text features and knowledge-based feature extraction to detect DGAs that randomly produced malicious domains. Because attackers only know how the DGA method works, he surveyed an algorithm to identify the DGA more efficiently. His model produced extremely excellent classification accuracy results. Finally, he suggested that by training and evaluating the dataset, he might improve the proposed technique.

From the above literature, we realize that researchers put their efforts into employing deep learning techniques and finding the best possible trade-off between privacy and utility because of the high processing power that these methods require.

3. Motivation

As it turned out, there are quite a few capable systems for categorizing DGAs, but their ability depends mainly on the dataset used for their training. Most of the published research uses DGA names that are publicly available and have resulted from published related projects and successful reverse-engineering efforts for various DGA families. The problem with these datasets is the limited number of names they consist of and, secondly, the large percentage of them that are obsolete [13]. Admittedly, they lack names from
more recent DGAs, reducing the discernment of trainee models regarding emerging DGAs. On the other hand, organizations specializing in cyber security and ISP providers train their models, using for the training process data that they are in no way willing to share due to competition and financial interests. The problem becomes more realistic if we consider that one of the largest sources of both DGA or legit names, a recursive DNS server, obeys strict privacy protocols, making it impossible to access essential data (e.g., logs with DNS queries) for the best and most up-to-date legit names, a recursive DNS server, obeys strict privacy if we consider that one of the largest sources of both DGA or legit names, a recursive DNS server, obeys strict privacy protocols, making it impossible to access essential data (e.g., logs with DNS queries) for the best and most up-to-date legit names, a recursive DNS server, obeys strict privacy protocols, making it impossible to access essential data (e.g., logs with DNS queries) for the best and most up-to-date.

So, we come up with a scenario in which, while all stakeholders want to train their models, using for the training process data with DNS logs with DNS queries) for the best and most up-to-date legit names, a recursive DNS server, obeys strict privacy protocols, making it impossible to access essential data (e.g., logs with DNS queries) for the best and most up-to-date legit names, a recursive DNS server, obeys strict privacy protocols, making it impossible to access essential data (e.g., logs with DNS queries) for the best and most up-to-date legit names, a recursive DNS server, obeys strict privacy protocols, making it impossible to access essential data (e.g., logs with DNS queries) for the best and most up-to-date.

The above situation seems to lead to a dead end. To avoid compromising privacy while promoting scientific research into large datasets to improve digital security, it is imperative to simultaneously implement technical solutions to address data protection and usage requirements [16, 17]. A serious answer to this situation is using secured multiparty computation (SMPC) [18] cryptography to train a deep learning model, combining all available data during training without disclosure and meeting the requirements for privacy [7, 13, 19].

SMPC is a cryptographic technique that allows different parties to perform calculations through inputs while keeping these inputs private [18, 20, 21]. Essentially, in this model, a set of parts with private inputs performs distributed functions, ensuring the required privacy and security issues. Conceptually, SMPC replaces a reliable intermediary to implement reliable calculations [21–25].

\[ L_{VAE} = L_{VAE_1} + L_{VAE_2} = \beta D_{KL}(p_{E_1}(z|\phi(x)) \parallel p(z)) - \mathbb{E}_{p_{E_1}(z|\phi(x))} \left[ \log p_{D_1}(\phi(x)|z) \right] + \beta D_{KL}(p_{E_2}(z|\pi(y)) \parallel p(z)) - \mathbb{E}_{p_{E_2}(z|\pi(y))} \left[ \log p_{D_2}(\pi(y)|z) \right]. \]

where \( \beta \) is the model hyperparameter, \( E_1 \) and \( D_1 \) are the encoder and decoder of DGA, and \( E_2 \) and \( D_2 \) are the encoder and decoder of the descriptions.

Siamese VAE encodes the characteristics \( \phi(x) \) in a probabilistic profile, which is modeled as standard, so it depends on an average value and a scatter table. Then a point of the standard integration space \( z \sim N(\mu, \Sigma) \) is sampled and decoded. The error is added to the total error function of the model so that [27, 32]

The MMD ensures that the DGAs and descriptions are adequately decoded and form areas in the embedding space, but their distributions are synchronized by minimizing the Wasserstein distance to which it applies as follows [29, 33, 34]:

\[ W_p(\mu, \nu) = \left( \inf_{\gamma \in \Gamma(\mu, \nu)} \int_{\mathbb{M} \times \mathbb{M}} d \gamma(x, y) \right)^{1/p}. \]

Finally, the synchronization of the distributions is calculated as follows:

\[ L_{DA} = \sqrt{\mu_1 - \mu_2^2 + \sum_1^{\frac{1}{2}} \sum_2^{\frac{1}{2}} 2 \text{Frobenius}}. \]

Although at this point the model works satisfactorily, the cross-synchronization technique is additionally used where

\[ L_{CA} = \phi(x) - D_1(E_2(\pi(y))) + \pi(y) - D_2(E_1(\phi(x))). \]

The VAE of the descriptions is required to decode DGAs, and the DGA decoder to decode descriptions, so the total error function of the model is

\[ L(x, y; E, D) = L_{VAE} + \gamma L_{CA} + \delta L_{DA}. \]
where γ, δ are hyperparameters of the model.

Finally, a simple Softmax classifier is used to classify embedding space in |Y| categories. An indicative architecture of the proposed system is shown in Figure 1.

To engage stakeholders who wish to enhance their models using the broadest possible variety and the most recent data available for education without the potential exposure of their private data, a machine learning protocol based on the SMPC technique is implemented [20, 22]. The proposed function offers participants the same possibility, as it allows the calculation of its value F only through the exchange of messages between n participants. Such a calculation could theoretically be performed in the presence of an inviolable and trustworthy referee other than n the participants, to whom each would give his value d_i, and he would correctly calculate its value F and announce to everyone only the result, as would be the case with the use of federated learning techniques.

In our case, we are interested in and present the implementation of a cumulative for calculating the function (d_1, . . . , d_n) = d_1 + d_2 + · · · + d_n based on the Shamir secret sharing (3S) method for n participants with n threshold. The 3S algorithm is based on the secure splitting and sharing of information between several participants. Each of them receives a value unrelated to the secret (in this case, the training data of the machine learning model is considered a secret), called a share of the secret, which has no utility. The secret can only be recreated if several parts of it are combined. For a total number of n shares to be defined, the minimum number t ≤ n is set initially, called the threshold (t, n) required to recover the secret S. t − 1 random integers a_1, a_2, . . . , a_{t−1} are selected while a_0 = S, to implement the following polynomial [18, 22, 35]:

\[ f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{t−1} x^{t−1}. \]  \tag{6}

Based on this polynomial, we obtain n random points (i, f (i)) : 0 ≠ i. Each point is communicated to one of the n participants. Having the polynomial (x), for the value x = 0, we get the value (0) = a_0, which is the secret S. It is noted that to maintain the correct secrecy, all operations are done with elements of a finite field F with size P where P first number, greater than all the coefficient values of the polynomial and the values t and n.

For any subset of t points, the above polynomial can be reconstructed using the Lagrange interpolation. Specifically, let n + 1 points (x_0, y_0), . . . , (x_n, y_n), where all x_j are different from each other. The Lagrange interpolation polynomial of P_n (x) degree ≤ n is given by the type as [22, 36]

\[ P_n(x) = l_0(x)f(x_0) + l_1(x)f(x_1) + \ldots + l_n(x)f(x_n) = \sum_{i=0}^{n} l_i(x)f(x_i). \]  \tag{7}

with:

\[ l_i(x) = \prod_{0 \leq j < i, j \neq i} \frac{x - x_j}{x_i - x_j}. \]  \tag{8}

The remainder can be bound as

\[ |R| \leq \max_{x_i \leq x \leq x_k} |f^{(k+1)}(\xi)|. \]  \tag{9}

The proposed protocol has the following steps [36, 37]:

1. Each participant p with a value d creates a random polynomial of n-th degree with a fixed value its hidden value d_p, as \( f(x) = d_p + a_1 x + a_2 x^2 + \ldots + a_n x^n \).

2. Calculates n values of (x) for n different but predefined values x_p, with x_p ≠ 0, one for each participant, including himself.

3. Sends to each participant p the corresponding value (x_p).

4. Steps 1 to 3 are performed by all participants, and each one sends the corresponding values of the random polynomial (x). An essential element of the process is that f_p values are not sent randomly. Assuming that each default value x_p is assigned to a specific participant p, then the corresponding value of (x_p) for the corresponding x must be sent to him by all participants.

5. Dataset

Two different sets of domain names were used to carry out the experiments. In the first dataset, 400,000 records are used from nonwordlist-based DGAs alone. Half of them came from Alexa’s collection of the top 1 million randomly selected from the most popular names, and the rest were created by running specific DGA algorithms. Ten different DGAs were executed in more detail, and 20,000 names were generated for each of the above algorithms. The second dataset uses wordlist-based DGAs and includes 500,000 records, half of which came from the Alexa collection of the top 1 million randomly selected from the most popular names [4, 6]. The rest were created by executing ten different wordlist-based DGAs. It should be noted that we
used domains that were related to education institutions (.ac, .edu, etc.), and e-learning software (Zoom, Cisco Webex, etc.)

An evaluation dataset consisting of 1,500,000 domain names was used to evaluate the method. Of these 800,000 legit domains, which are different from those of the training dataset, 550,000 come from the Alexa top 1 million collections, while domain name registration machines retrieved the rest. Respectively, of the 700,000 DGA domains, 300,000 are real DGAs domains registered in corresponding lists of security organizations such as BlackHoleDNS, while the remaining 400,000 have been created after executing 40 different DGAs. The following Figure 2 illustrates the dataset.

The 20 algorithms have not been included in the training dataset to make the dataset more realistic. However, even for those included, different seeds and wordlists were used in their execution, so that the domain names generated were different from those used for training. For the creation of DGA domains, the length of the domains was random, ranging from 6 to 21 alphanumeric characters written in the Latin alphabet. An entropy algorithm was also applied to the creation of the domains as a degree of uncertainty to enhance the realism of the generated domains. Even with the Alexa grams technique, the degree of sequence between the generated DGA domain and the list of domains derived from Alexa was calculated using the technique of the probabilistic model for the prediction of the next n-gram element. Finally, the word-gram process was used to calculate the degree of correlation - sequence between the DGA domain and 500,000 widely used words to predict the next word-gram element.

The metadata used for the second VAE was the end of the domain name, the degree of entropy of each domain (entropy), the degree of the sequence of each domain (n-gram), and the degree of correlation-sequence of each domain (word-gram) [32, 34].

Overall, Table 1 presents the datasets used in this study as follows.

#### 6. Experiments

To evaluate the performance of the proposed system per class as well as the estimation of the actual error during the training, we used the following measures [26, 38]:

| Metric       | Formula                                                                 |
|--------------|-------------------------------------------------------------------------|
| Sensitivity  | $\frac{tp}{tp + fn}$                                                  |
| Specificity  | $\frac{tn}{tn + fp}$                                                  |
| Precision    | $\frac{tp}{tp + fp}$                                                  |
| F1-score     | $\frac{2\times tp}{2\times tp + fn + fp}$                              |
| Rec. accuracy| $\frac{tp + f_n}{tp + fn}$                                            |

Where $tp$ = true positive, $fp$ = false positive, $fn$ = false negative, and $tn$ = true negative.

We conducted three experiments where, for the first time, training was performed with the nonwordlist-based DGA dataset and a test with the MixTest nonwordlist and wordlist DGA. Then training was done with the wordlist-based DGA dataset and testing was done with the MixTest nonwordlist and wordlist DGA. Finally, the two training datasets (nonwordlist-based DGA and wordlist-based DGA dataset) were combined, and the MixTrain nonwordlist was created and the wordlist DGA, which was tested with the MixTest nonwordlist and wordlist DGA. The following Table 2 shows the results of the followed procedure.

As can be seen from the table above, the generalizability of the system is significantly enhanced by the MixTrain (nonwordlist and wordlist-based DGA) dataset, which includes many more and much more complete samples of DGA domains.

In the case of applying the SMPC algorithm, we proved its functionality by proving that ring uniformities “retain” operations. Specifically, we demonstrated that there exists an isomorphism of rings of polynomials (i.e., be 1–1) where at least $R$ is a transposition ring, and $I$ is an ideal of $R$ with [18, 25, 35]

$$R[t] \rightarrow R(t)[t].$$

In detail, considering the illustration as follows:
\[
\Phi: \mathbb{R}[t] 
\rightarrow \left( \frac{\mathbb{R}}{I} \right)[t], \quad P(t) = \sum_{k=0}^{\infty} a_k t^k \rightarrow \Phi(P(t)) = \sum_{k=0}^{\infty} (a_k + I) t^k. \tag{11}\]

Then \(\Phi(1_{R/I}) = \Phi(1) = 1 + I = 1_{R/I},\) i.e., the illustration \(\Phi\) sends the unit of \(R[t]\) to its unit \((R/I)[t]\).

If \(P(t) = \sum_{k=0}^{\infty} a_k t^k\) and \(Q(t) = \sum_{k=0}^{\infty} b_k t^k\) are two polynomials in the ring \(R[t]\), then we can assume without harm to the generality that \(n \leq m\) and then we can write
\[
P(t) + Q(t) = \Phi \left( \sum_{k=0}^{m} a_k t^k + \sum_{k=0}^{m} b_k t^k \right) = \Phi \left( \sum_{k=0}^{m} (a_k + b_k) t^k \right) = \sum_{k=0}^{m} (a_k + b_k + I) t^k = \sum_{k=0}^{m} a_k t^k + \sum_{k=0}^{m} b_k t^k + \sum_{k=0}^{m} I t^k = \Phi(P(t)) + \Phi(Q(t)). \tag{12}\]

Similarly, setting \(a_r = \sum_{k=0}^{r} a_k t^k, 0 \leq k \leq n + m\) we will have

**Figure 2:** DGA domains unique client IPs and DNS queries by https://data.netlab.360.com/dga/.

**Table 1:** Training and test datasets.

| Training datasets               | Test dataset                          |
|---------------------------------|---------------------------------------|
| Nonwordlist-based DGA 200,000 legit | MixTest nonwordlist and wordlist DGA 800,000 legit |
| 200,000 DGA                     | 250,000 DGA                           |

| Training dataset               | Accuracy | Recall | Precision | F-score |
|--------------------------------|----------|--------|-----------|---------|
| Nonwordlist-based DGA          | 0.8949   | 0.8883 | 0.8904    | 0.8903  |
| Wordlist-based DGA             | 0.9072   | 0.9038 | 0.9056    | 0.9058  |
| MixTrain (nonwordlist & wordlist-based DGA) | 0.9260   | 0.9263 | 0.9259    | 0.9261  |
\[ \Phi \left( P(t) \cdot Q(t) \right) = \Phi \left( \sum_{k=0}^{m} a_k t^k \cdot \sum_{k=0}^{m} b_k t^k \right) = \Phi \left( \sum_{k=0}^{n} \left( c_k + I \right) t^k \right) = \sum_{k=0}^{n} \left( a_k + I \right) \left( b_{k-1} + I \right) t^k \]

Thus, the illustration \( \Phi \) is a homomorphism of rings which in addition is a homomorphism because if \( A(t) = \sum_{k=0}^{n} (a_k + I) t^k \) is a typical ring element \((R/I)[t]\), then setting \( P(t) = \sum_{k=0}^{n} a_k t^k \in R[t] \) we will have \( \Phi(P(t)) = A(t) \).

Let \( P(t) = \sum_{k=0}^{n} a_k t^k \in \text{Ker}(\Phi) \), then \( \Phi(P(t)) = \sum_{k=0}^{n} (a_k + I) t^k = 0_{(R/I)[t]} = I \) is the zero polynomial in the ring \((R/I)[t]\), i.e., \( a_k + I = I \), and therefore \( a_k \in I \), \( 0 \leq k \leq n \). This means that the polynomial \( P(t) \in I[t] \). Conversely, if \( P(t) \in I[t] \), then \( v_k \in I \), \( 0 \leq k \leq n \), and then obviously \([21, 23]\)

\[ \Phi(P(t)) = \sum_{k=0}^{n} \left( a_k + I \right) t^k = \sum_{k=0}^{n} \left( 0_{R/I} \right) t^k = 0_{(R/I)[t]} \]  

Therefore

\[ \text{Ker}(\Phi) = I[t]. \]  

and so, the subset \( I[t] \) is an ideal of \( R[t] \) as the nucleus of a ring homomorphism. Finally, because the imaging \( \Phi \) is training, it follows that \( \Phi \) induces the ring isomorphism of the original hypothesis as

\[ \frac{R[t]}{I[t]} \cong \left( \frac{R}{I} \right)[t]. \]

We can therefore use the methodology of the SMPC algorithm to apply machine learning techniques with very high performance even in cases of encrypted data.

7. Discussion and Conclusions

The detection and timely assessment of DGA domains and DGA network traffic allows for the detection of incidents and the corresponding identification of correlations and relationships with security incidents, significantly mitigating the effects of sophisticated cyber-attacks. Individual efforts by independent actors cannot perform effectively and quickly in the field of knowledge discovery. On the contrary, collaborative efforts, which, as it turns out, can be implemented with remarkable learning models that also work on encrypted data, can lead to a significant increase in the accuracy of results and the generalization of learning models. Also, the increasing nature of the data requires the rise of training datasets, always considering the adaptation of the method to the available memory resources and computing power.

Considering the need for realistic and accurate security incident detection systems, this paper presented an innovative and highly practical privacy-preserved machine learning methodology for the timely detection of DGA domains and the network traffic they generate, with respect to distance education functionality. This methodology combines the Siamese variational-autoencoder in a complete framework. It is a robust system that calculates the number of maximum probable intervals within which an event is likely to occur based on a parametric evaluation that uses realistic datasets.

An essential advantage of the method, which has been proven experimentally, is that VAEs can, by receiving a combination of data and metadata, detect complex and sophisticated DGA domains. The dynamic identification of the proposed system directly integrates all the information in the sequence of the sample set, creating conditions for a realistic approach in recognizing security events.

Significant improvements in the evolution of the proposed system mainly concern the optimization of VAE hyperparameters, which are sensitive to modifications in determining the input data trend. Also, a significant improvement involves how the system is investigated with dynamic variational inference methodologies to provide a detailed approach to the subsequent probability of unobserved variables and apply a statistical conclusion for these variables.[39].

Data Availability

The data used in this study are available from the author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This study was supported by the Project of Exploration on the Training Model of Primary School English Teachers in the Context of Professionalism (No. 2016-JSJYYB-207).

References

[1] B. Paris, R. Reynolds, and C. McGowan, “Sins of omission: Critical informatics perspectives on privacy in e-learning systems in higher education,” Journal of the Association for Information Science and Technology, 2021.
[2] D. Turnbull, R. Chugh, and J. Luck, “Transitioning to E-learning during the COVID-19 pandemic: how have higher education institutions responded to the challenge?” Education and Information Technologies, vol. 26, no. 5, pp. 6401–6419, 2021.
[3] P. Karunakaran, “Deep learning approach to DGA classification for effective cyber security.” J. Ubiquitous Comput. Commun. Technol. vol. 2, no. 4, pp. 203–213, 2021.
[4] K. Demertzis and L. Iliadis, “Evolving Smart URL Filter in a Zone-Based Policy Firewall for Detecting Algorithmically
Generated Malicious Domains,” in *Proceedings of the Statistical Learning and Data Sciences*, pp. 223–233, SLDS, London, UK, April 2015.

[5] K. Shaukat, S. Luo, V. Varadharan et al., “Performance comparison and current challenges of using machine learning techniques in cybersecurity,” *Energies*, vol. 13, no. 10, p. 2509, 2020.

[6] T. Chin, K. Xiong, C. Hu, and Y. Li, “A machine learning framework for studying domain generation algorithm (DGA)-Based malware,” in *Proceedings of the Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol. 254, Rochester, USA, December 2018.

[7] J. Gawlikowski, C. R. N. Tassi, M. Ali et al., “A Survey of Uncertainty in Deep Neural Networks,” 2021, http://arxiv.org/abs/2107.03342.

[8] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, “Deep learning for anomaly detection,” *ACM Computing Surveys*, vol. 54, no. 2, pp. 1–38, 2022.

[9] A. Klasič-Miličević and M. Ivanović, “E-learning personalization systems and sustainable education,” *Sustainability*, vol. 13, no. 12, p. 6713, 2021.

[10] D. Zhang, X. Chen, D. Wang, and J. Shi, “A survey on collaborative deep learning and privacy-preserving,” in *Proceedings of the 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, pp. 652–658, IEEE, Guangzhou, Jun, June-2018.

[11] T. Ryffel, A. Trask, M. Dahl et al., “A generic framework for privacy preserving deep learning,” 2018.

[12] B. Chang, “Student privacy issues in online learning environments,” *Distance Education*, vol. 42, no. 1, pp. 55–69, 2021.

[13] L. Alzubaidi, J. Zhang, A. J. Humaidi et al., “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions,” *Journal of Big Data*, vol. 8, no. 1, p. 53, 2021.

[14] Z. Yu, A. M. Abdulghani, A. Zahid, H. Heidari, M. A. Imran, and Q. H. Abbasi, “An overview of neuromorphic computing for artificial intelligence enabled hardware-based hopfield neural network,” *IEEE Access*, vol. 8, pp. 67085–67099, 2020.

[15] K. Al Jallad, M. Aljindi, and M. S. Desouki, “Anomaly detection optimization using big data and deep learning to reduce false-positive,” *Journal of Big Data*, vol. 7, no. 1, p. 68, 2020.

[16] T. Alshalali, K. M’Bale, and D. Josyula, “Security and privacy of electronic health records sharing using hyperledger fabric,” in *Proceedings of the 2016 International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 760–763, IEEE, Las Vegas, NV, USA, Sep. 2018.

[17] M. Dhingra, M. Jain, and R. S. Jadon, “Role of artificial intelligence in enterprise information security: a review,” in *Proceedings of the 2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, pp. 188–191, IEEE, Waknaghat, Dec. 2016.

[18] C. Zhao, S. Zhao, M. Zhao et al., “Secure multi-party computation: theory, practice and applications,” *Information Sciences*, vol. 476, pp. 357–372, 2019.

[19] A. Darwish, “Bio-inspired computing: algorithms review, deep analysis, and the scope of applications,” *Future Computing and Informatics Journal*, vol. 3, no. 2, pp. 231–246, 2018.

[20] J. Bringer, H. Chabanne, and A. Patey, “Privacy-preserving biometric identification using secure multiparty computation: an overview and recent trends,” *IEEE Signal Processing Magazine*, vol. 30, no. 2, pp. 42–52, 2013.

[21] B. Jiang, “Two-party secure computation for any polynomial function on ciphertexts under different secret keys,” *Security and Communication Networks*, vol. 2021, pp. 1–7, 2021.

[22] H. Akbari-Nodehi and M. A. Maddah-Ali, “Secure coded multi-party computation for massive matrix operations,” *IEEE Transactions on Information Theory*, vol. 67, no. 4, pp. 2379–2398, 2021.

[23] R. Hou, F. Tang, S. Liang, and G. Ling, “Multi-party verifiable privacy-preserving federated k-means clustering in outsourced environment,” *Security and Communication Networks*, vol. 2021, pp. 1–11, Article ID 3630312, 2021.

[24] S. Zavvak and M. Iskifyeli, “Anomaly-based intrusion detection from network flow features using variational autoencoder,” *IEEE Access*, vol. 8, Article ID 108346, 2020.

[25] M. Sepehri, S. Cimato, and E. Damiani, “Privacy-preserving query processing by multi-party computation,” *The Computer Journal*, vol. 58, no. 10, pp. 2195–2212, 2015.

[26] T. W. Anderson, “An introduction to multivariate statistical analysis,” Wiley, NY, London, Sydney, 2003.

[27] D. P. Kingma and M. Welling, “An introduction to variational autoencoders,” *Foundations and Trends in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019.

[28] Y. Bian and X. Tang, “Abnormal detection in big data video with an improved autoencoder,” *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1–6, Article ID e9861533, 2021.

[29] Z. Gu and Y. Yang, “Detecting malicious model updates from federated learning on conditional variational autoencoder,” in *Proceedings of the 2021 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 671–680, IEEE, Portland, OR, USA, May 2021.

[30] E. Protopapadakis, A. Voulodimos, A. Doulamis, N. Doulamis, D. Dres, and M. Bimpas, “Stacked autoencoders for outlier detection in over-the-horizon radar signals,” *Computational Intelligence and Neuroscience*, vol. 2017, pp. 1–11, Article ID e5891417, 2017.

[31] X. Xu, J. Li, Y. Yang, and F. Shen, “Toward effective intrusion detection using log-cosh conditional variational autoencoder,” *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6187–6196, 2021.

[32] C. Doersch, “Tutorial on Variational Autoencoders,” 2021, http://arxiv.org/abs/1606.05908.

[33] M. M. Abdelsamea, G. Gnecco, M. M. Gaber, and E. Elyan, “On the relationship between variational level set-based and SOM-based active contours,” *Computational Intelligence and Neuroscience*, vol. 2015, pp. 1–19, Article ID e109029, 2015.

[34] D. P. Kingma and M. Welling, “Auto-Encoding Variational Bayes,” 2014, http://arxiv.org/abs/1312.6114.

[35] Y. Sun, Q. Wen, Y. Zhang, and W. Li, “Privacy-preserving self-helped medical diagnosis scheme based on secure two-party computation in wireless sensor networks,” *Computational and Mathematical Methods in Medicine*, vol. 2014, pp. 1–9, Article ID e214841, 2014.

[36] Z. Wang, S.-C. S. Cheung, and Y. Luo, “Information-theoretic secure multi-party computation with collusion deterrence,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 980–995, 2017.

[37] L. Zhou, L. Wang, Y. Sun, and T. Ai, “AntNest: fully non-interactive secure multi-party computation,” *IEEE Access*, vol. 6, Article ID 75639, 2018.

[38] S. Raschka, “An Overview of General Performance Metrics of Binary Classifier Systems,” 2014, http://arxiv.org/abs/1410.5330.

[39] R. Beyah, B. Chang, Y. Li, and S. Zhu, *Springer International Publishing*, Springer, Berlin, Germany, pp. 433–448, 2018.