International Journal of Data Informatics and Intelligent Computing (IJDIIC)
Vol. 2, No. 4, December 2023, pp. 20~32
DOI: 10.59461/ijdiic.v2i4.90

Consideration of Data Security and Privacy Using Machine Learning Techniques

Thanh Chi Phan[1], Hung Chi Tran[2]
[1]Quang Tri Teacher Training College, Hanoi University of Science and Technology, Vietnam
[2]Information Technology Engineer, IT Center, Quang Tri Teacher Training College, Dong Ha City, Quang Tri Province, Vietnam

Article Info

Article history:

Received November 07, 2023
Revised December 01, 2023
Accepted December 05, 2023

Keywords:

Machine learning
Security
Cryptography
Privacy-preserving data protocol

ABSTRACT

As artificial intelligence becomes more and more prevalent, machine learning algorithms are being used in a wider range of domains. Big data and processing power, which are typically gathered via crowdsourcing and acquired online, are essential for the effectiveness of machine learning. Sensitive and private data, such as ID numbers, personal mobile phone numbers, and medical records, are frequently included in the data acquired for machine learning training. A significant issue is how to effectively and cheaply protect sensitive private data. With this type of issue in mind, this article first discusses the privacy dilemma in machine learning and how it might be exploited before summarizing the features and techniques for protecting privacy in machine learning algorithms. Next, the combination of a network of convolutional neural networks and a different secure privacy approach is suggested to improve the accuracy of classification of the various algorithms that employ noise to safeguard privacy. This approach can acquire each layer's privacy budget of a neural network and completely incorporates the properties of Gaussian distribution and difference. Lastly, the Gaussian noise scale is set, and the sensitive information in the data is preserved by using the gradient value of a stochastic gradient descent technique. The experimental results showed that a balance of better accuracy of 99.05% between the accessibility and privacy protection of the training data set could be achieved by modifying the depth differential privacy model's parameters depending on variations in private information in the data.

Corresponding Author:

Thanh Chi Phan
Quang Tri Teacher Training College
Hanoi University of Science and Technology
Vietnam
Email: thanhpc.sp@gmail.com

1. INTRODUCTION

Currently, the primary focus of artificial intelligence is data, and both domestically and internationally, there is a growing interest in protecting personal data. Unauthorized use of private, sensitive information, whether on the internet or offline, is punishable by law. In 2018, the European Union put forth explicit guidelines regarding the management of personal data that businesses gather. Companies are no longer allowed to gather, distribute, or analyze user data without consent. The implementation of protecting privacy in machine learning requires taking into account the peculiarities of machine learning in addition to employing regulations to restrict information leakage [1]. The primary assumption is to guarantee that private and sensitive data won't be either provided or received by unauthorized individuals throughout the training process.

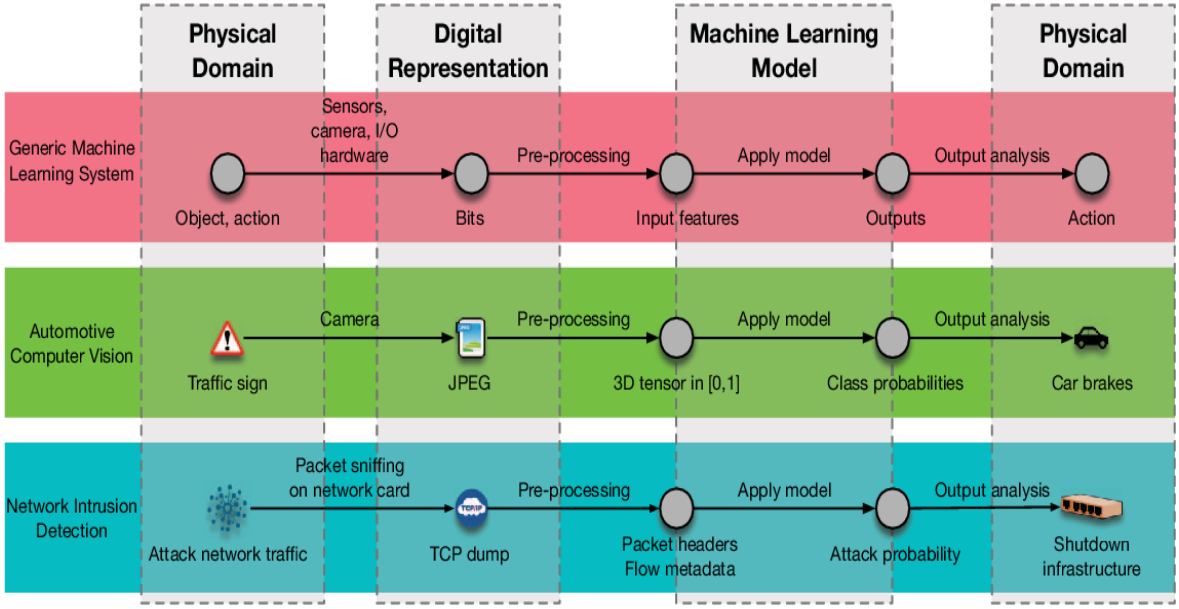

Figure 1. A computer vision model is used by an automotive system to recognize network intrusion detection systems [2].

Traditional neural network training involves data collectors gathering all parties' data, which is subsequently taught by centralized learning, as Figure 1 illustrates. As in the case of mobile application developers, the information collector and the data analyst might work together [2]. Additionally, it might be multiparty, for example, when developers exchange data with different data analytics firms. It is evident that in the centralized learning mode, users find it challenging to maintain possession of the data once it has been gathered, and they are also unsure of where and how best to use the data. Currently, some researchers are attempting to develop a global model using only local data. Google's 2017 proposal for federated learning serves as a typical example of this approach [3]. Federated learning provides users with ownership over their personal information, but it is not impenetrable protection against privacy threats.

Currently, different privacy algorithms and their many refinement strategies are the major means by which machine learning privacy protection is achieved. Researchers primarily focus on three components of differential privacy improvement: gradients, operation, and label, correspondingly [4]. The fundamental idea behind all differential privacy algorithms is to introduce noise into machine learning processes in a variety of ways and tactics to disrupt the neural network's ability to recall actual training data. A DP-SGD technique was proposed for this type of method, which does not support the convergence of complicated models [5]. The developed DP-GAN approach calculates the gradient from the Wasserstein once distance by using private information in noise protection data. This method works poorly in complicated datasets because it depends on generators to produce excellent training points [6]. In order to offer the best perturbation parameters, they integrated a deep encoder with multiple privacy approaches and then used a back-propagation algorithm to fine-tune the model parameters [7]. The global hypersensitivity computing layer was added to the encoder. Without fitting training data, this technique adds an extra network layer and the detrimental impact is especially pronounced in complicated datasets. They suggested a novel differential privacy protection method, called ADLM, as an enhanced technique. The predictive ability of the model is increased to 84.8% when the technique is adapted to the CIFAR-10 dataset, which is 14 percent greater than that of the DP-SGD approach, although it is still not optimal [8]. They developed a semi-supervised train of numerous teacher models using disjoint datasets and then aggregated the predictions [9]. This training approach has good security for privacy capabilities and can yield a reasonably high model accuracy. However, since the accuracy of the teacher model's prediction parameters determines how well the student model is trained, making multiple teacher models exceptionally accurate necessitates substantial input training, which is obviously devastating. Creating noise in the aggregation process for various datasets is likewise a challenging undertaking.

Large data sets can be automatically analyzed using machine learning, which generates models or decision-making processes that represent common patterns in the data [5]. Three groups of machine learning approaches are typically recognized based on the type of data that can be analyzed. Supervised learning: Supervised learning techniques are methods that are provided with examples of instruction in a format of inputs labeled with associated outputs [6][7] and machine translation [8]. The method's task is unsupervised when it receives unlabeled inputs. This covers issues like decreasing the dimensionality [9], model pre-

training, and clustering [10] points based on a similarity metric [11]. Clustering can be used, for example, in anomaly detection [12]. Reinforcement learning: Sequences of measures, observations, and incentives (such as video game runs) are examples of data that can be used in reinforcement learning (RL) [13], [14]. Planning and control are the focus of the ML area of RL, which aims to create a policy for behavior in the environment. RL agents discover things about their surroundings via experimentation. Recently, a machine beat a human expert in the game of Go thanks to reinforcement learning combined with supervised and unsupervised techniques [15].

This research focuses on the objectives of adversaries and potential protection and attack capabilities specific to machine learning-based systems [16]. The latter type of attack may affect the data source's privacy, particularly if model operators are not trusted. This is especially true when training medical diagnostic models with extremely sensitive patient clinical data. They are frequently carried out by altering the data that the machine learning system learns from or makes predictions from. These attacks come into the availability category when those adversarial behaviors try to deny authorized users access to the system's features or significant model results. Examining security and privacy from the standpoint of machine learning pipeline attacks and responses is the second viewpoint. Here, we examine the entire lifetime of a machine learning system, from training to speculation, and pinpoint the objectives and strategies of the adversaries at every stage [17].

While adding noise advances in speech recognition, natural language processing, target identification, and other fields, intelligent data recognition machinery is today more sophisticated than it has ever been. A rising number of academics are interested in deep learning because it is the most developed technique for intelligent data recognition [18]. The key characteristic of deep learning is its ability to use neural networks to combine low-level information into higher-level, more abstract features. This allows for the discovery of dispersed feature representations in data, which significantly improves the differential privacy model's prediction accuracy. The features of Gaussian distribution and differential privacy can be effectively combined using this approach. It can better safeguard confidential details in the data collection and offers greater data availability in combination with privacy protection. Reconstructing training from small sample training is possible, and the attack impact is much reduced with larger sample sizes. Equation solving is the main technique used in the initial model theft attempt, and it is only suitable for basic linear binary models [19].

The research findings are summarized as follows: The backdrop is covered in Section 2, while the method is covered in detail in Section 3. Results and discussion in Section 4. The article is finally concluded in Section 5.

2. LITERATURE REVIEW

This section delves further into the privacy concerns surrounding machine learning and its application to privacy protection. Sensitive information is an issue that arises as gathering and analyzing data advances. There are two primary ways that machine learning privacy leaks: (1) Mass data collecting is the cause of direct privacy violations. Unreliable data collectors unlawfully exchange, transfer, and gather personal data without the consent of the subjects. (2) Inadequate model generalization capacity leading to indirect privacy disclosure. It mostly shows up as the ability of an untrustworthy data analyst to interact with the model and chat about specific sensitive qualities in training data that are unknown [20]. The fundamental source of this issue is that data has a stronger "memory" capacity when a model is trained with increased complexity; hence, models operate differently and are extremely sensitive to variations in the data.

The main objective is referred to as indirect privacy disclosure. Attacks on privacy primarily happen during the model application stage. Attackers can only deduce pertinent information because they are unable to obtain training data directly. An attacker can be completely unaware of the algorithm and data, or they might be somewhat aware of it thanks to things like known types of models or data features. Refactoring and member-based reasoning are the two primary methods of privacy assault, depending on the attacker's objective [21].

The term "reconstruction attack" describes an attacker's effort to piece together private data or the target model of a particular person using training data. The term "model stealing attack" refers to the latter, whereas "model inversion attack" refers to the former. For models of machine learning with basic structures, dynamic analysis or estimating similarity between records can be used to forecast personalized medicine linear prediction models, provided that the patient's fundamental data and prediction outcomes are available. Additional information, such as sample labels, is used for sophisticated deep-learning models. The confidence process is employed to discern artificially generated virtual portraits, thereby reinstating the authenticity of tiny training samples, and the effect of attacks will be significantly reduced with high sample sizes. In terms of model theft attacks, only basic linear binary models can benefit from the equation-solving method used in the initial model theft assault [22]. Predictive confidence allows for a large improvement in the attack effect and allows the strategy for a decision tree model to be proposed by certain academics. An

inversion attack intended to obtain an alternative model can greatly increase, even though an attack aiming at stealing an algorithm has little or no interest in data. Machine learning algorithms are crucial property rights for businesses in real-world application settings. Businesses will suffer significant damages if they are stolen [23].

An attack known as a "member inference attack" involves will be utilized in the development of the model. When a diagnostic model is developed using AIDS patient data, for example, inferential attacks may have dangerous repercussions. The mode uses simulated data to create the shadow model, which is an alternative description in "black box" mode. Concurrently, the model for the attack was programmed to assess the target model's learning impact based on the variation in the output. The foregoing restrictions are loosened in the ensuing research, and a more comprehensive attack model is suggested—an attack in the person who attacked can determine the target model's average loss and determine whether the information being attacked constitutes training data by calculating the model's data loss [24].

As a result, current machine learning privacy attacks are very limited, and they are only effective in specific situations [25]. Given that these attacks frequently result in the death of sophisticated models, it is imperative that we investigate these matters. However, researchers should take into account privacy issues; machine learning is subject to a variety of security risks. The primary distinction between a security issue and a privacy issue is that while the former results in the direct or indirect disclosure of training information, the model remains unaffected; on the other hand, the latter will lead to the malicious induction or destruction of the model's internal logic, preventing the model from performing as intended. Both the model training and the model application phases are susceptible to malicious attacks against machine learning, the most common of which are antisample and poisoning attacks. These days, machine learning also faces security challenges [26].

A machine learning scenario that raises the possibility of privacy disclosure and calls for privacy protection steps is known as a privacy protection scenario. Different privacy settings call for different privacy protection systems. The foundation of creating privacy protection schemes is an understanding of privacy protection scenarios [27].

Prediction comes after machine learning, and training comes before it. Different privacy issues arise at different phases of machine learning, and because machine learning is a technology, the protection strategies used also vary. Nowadays, deep neural networks almost always use homomorphic encryption during the prediction phase and very never computationally expensive technology that requires a lot of computing power and network infrastructure [28]. Its communication and computing requirements are significant, and it still requires high throughput hardware support in the absence of encryption. Thus, deep neural system prediction typically uses homomorphic encryption. Nonetheless, the effective machine learning privacy protection technique based on technology for encryption in training for research is still. Model training techniques based on machine learning are joint, distributed, and centralized. The concentrated model of learning does not necessarily centralize each participant's training data to a single server when using dispersed learning. Each participant may receive training data in a horizontal, vertical, or random fashion [29]. Joint learning can keep the instructional data dispersed by first creating a collective model and then training concurrently using data from several users. Though federated learning faces a more difficult learning environment, it has garnered interest from academic and industry circles because it places a higher value on the security of user data privacy.

Three strategies are frequently employed to safeguard private information related to machine learning: secure multiparty systems computing for privacy protection. Differential encryption technology is one of them. It is a data distortion technique that uses manual intervention or noise addition to guarantee secrecy and encryption that is homomorphic. Techniques for multiparty computing are within the category of cryptography, which uses security standards to safeguard data privacy while it is being used. The aforementioned techniques are frequently combined, such as when homomorphic encryption is used in conjunction with secure multiparty computing or when many privacy-secure multiparty computing techniques are applied simultaneously.

3. METHOD

This section describes how the IoT architecture has improved communication security. SDN establishes a safe channel of communication between network objects in CNN. In this instance, there are several subnets within the network topology. Because each subnet's members will have very comparable positions and patterns of movement, network topology communication will always be stable. Additionally, a controller node is tasked with overseeing and verifying the communication between each subnet's participants in this topology. Network traffic is also monitored by a learning model in this manner to recognize security threats and assaults within its subnet.

Network nodes have heterogeneous communication characteristics because wireless networks use radio equipment made of diverse technologies. The presumed network is, hence, non-homogeneous. Since

the anticipated network structure is built by measuring the intensity of a radio signal that each node receives, the distance that exists between two nodes may be determined. Therefore, by determining the signal strength that was received from the nearby nodes, network equipment without GPS or a global positioning system can determine their distance from one another. A learning model with the ability to gather and analyze data traffic is installed on every control node in the SDN [30].

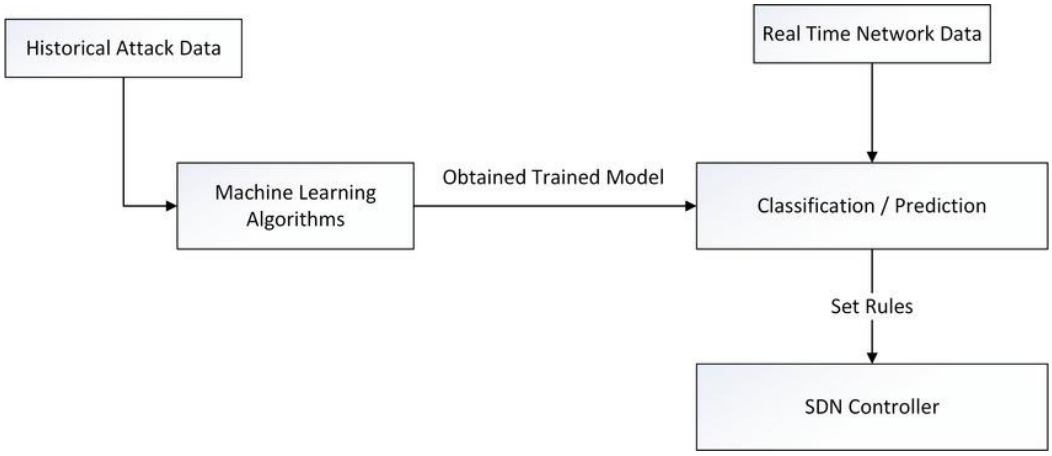

Figure 2. Proposed Block diagram of this work

Figure 2 depicts the specifics of the CNN steps as a diagram. This picture illustrates how CNN is repeated at predetermined intervals of time. Each controller in CNN will give the other controllers access to a list of verified users associated with its subdomain. In this manner, messages are exchanged between the controllers to establish user credit in the event that contact between two users is required. Data routing will be completed if at least one controller has authenticated each of both sides of the communication. The Prim algorithm and the minimal spanning tree are used to regulate the network topology. In this step, minimum-spanning trees are built by each node, forming the local topology of the network. Then, a structure with levels is created for data routing by leveling nodes of the network and weighing network connections. Ultimately, the orderly tree structure is used to route the data to its destination. The controlling node of a subnet serves as the intermediary for all node traffic linked to that subnet, according to the topology suggested by this research. As a result, every controller node continuously analyzes network traffic data to spot assaults using an EL learning model. This model, which consists of three learning models, determines whether assaults might be present in a given traffic flow by analyzing statistical data that is taken from each one.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{\frac{x^2+y^2}{2\pi^2}} \quad (1)$$

The secure communication architecture between network objects will be determined by creating a topology structure in the first stage of CNN. In order to accomplish this, it is first required to determine each active node's neighbor list within the network. This is accomplished by trading Hello control messages. Through the broadcasting of this message, each node in the process notifies the other nodes of its presence. It does this by storing its distinctive identification number in the material of the control packet. The node measures and logs the signal intensity it receives from every neighboring node during these exchanges. To find and eliminate the poor network connections, the network components exchange neighbor lists in the following stage. Each operating node will communicate in order to accomplish this.

$$x_j = \frac{1}{1+e^{net_j}} \quad (2)$$

$$E = \frac{1}{2}\sum_j (x_j - t_j)^2 \quad (3)$$

Active node A determines the average signal strength on each end of the connection using the formula RAVG = RSSIAB + RSSSIBA 2, taking into account the intensity of the signal it received over the relationship between A and B. This technique can somewhat lessen the detrimental impact of noise on signal evaluation. When the threshold, P, is exceeded by the average level of signal strength, RAVG, the link between nodes A and B is deemed active since it has adequate quality. In the absence of it, active sites A and B delete one another from the list of nearby ones if their connection lacks the necessary quality.

Each network node that implements this method creates a set of suitable-quality communication links amongst data that is delivered to the neighbors with the highest quality of neighborliness upon transmission of the topology development control signal by each nearby node. The node in a network with the highest grade of neighborhood receives the topology development control packets if that procedure is repeated. The central node, or Ct, is this node. Consequently, choosing a node to serve as the network architecture's central node is the initial stage in creating a hierarchical tree topology. To identify the topology center, the neighborhood degree feature may be a useful characteristic. By broadcasting all of the control packets, the controller nodes identified in the prior phase in CNN first identify their neighbors. When the structure construction packet is distributed and identified, it will be designated as the topologies center. Each responding node records the congestion, vitality, and estimated distance of the control message during this process, saves it in a response packet, and then transmits it to the sender node.

Secure data routing will be implemented using this structure following the construction of the hierarchical tree topology. It is evident from the tree topology that it has only a single path connecting the two subdomains. However, the domain controllers of each subdomain must exchange member data in order to ensure private information routing between nodes that travel in the network. In this manner, before a node transfers, it notifies the controllers regarding additional subdomains by packets, including the destination node's ID. Via the central node Ct, the controller with the destination node within its subdomain transmits the message of confirmation to the source node. The connection between the two nodes will be created in this manner.

. While node B is normal, node A transmits malicious messages. All of these nodes are presumed to be communicating with one another. As previously indicated, every network node exchanges data via its subnet controller, which uses a model of a neural network to verify every message delivered by a subnet member. The message will be banned and erased if the computerized neural network classifies the incoming message as an attack. In the hypothetical case of node A transmitting a message to node B, this condition was met. Conversely, the message from node B reaches the receiver node A because the neural network in the controller recognizes it as normal. The procedure for identifying assaults using an artificial neural network is described below. Standardizing packet traffic data is the initial stage in the technique of identifying assaults. The following procedures are followed in order to standardize the data: The flow of traffic being processed has nominal properties that are assigned numerical values. For instance, the values one through three can be substituted for the ICMP, UDP, and TCP states that might be assigned to the "connection type" attribute.

The amalgamation of "artificial neural network," "K nearest neighbor," and then "support vector machine" has been utilized to perceive assaults using the generated structures after standardizing the traffic flow data. Training samples are used to train each of the learning models that have been discussed separately. Each of those learning models processes the test samples, which are network traffic features, and each model's output is defined as a logical variable. In this instance, each learning model's True output indicates the presence of an outcome subsequent to the result of the three models of learning being determined. In this instance, every test sample will be a member of the group of outputs whose label receives the most votes from the learning models. Put differently, if at least two of the system's learning models identify a certain traffic flow’s characteristic as an incursion, the suggested aggregate system will identify that movement of traffic as an attack.

4. RESULTS AND DISCUSSION

The results of applying neural networks to encrypted statistics are shown in this section. For implementation, we used HELib [31]. All calculations were performed on a virtual computer running Ubuntu 14.04, a Core i5 processor, and 8GB of RAM. Initially, we use encrypted data to train the models, and we track how long the training process takes. Next, we categorize encrypted instances using the trained model, and we track both the model's running time and accuracy. In three datasets, Fundoscopy, Chest X-ray, and Dermoscopy, diagnosis results can be arbitrarily manipulated by adversarial attacks. It presents the mean execution duration over several algorithm iterations, with a standard variation of less than 3.0% across the board. During computations, we should also take into account the increasing noise in the ciphertexts. Bootstrapping is one method of handling the noise, but it requires a lot of processing. We employ a different strategy to deal with this issue: the client receives the ciphertext from the server, decrypts or re-encrypts it, and the value of L determines how many operations on the ciphertext are permitted. Higher values of L need fewer communications between the client and the server than smaller ones, which necessitate more communications.

Table 1. Comparison of different Hidden Layers with Network creations and Backpropagation [32].

Hidden Layer (s)	Network Creation (s)	Backpropagations (s)
1	41.32	108.82
2	55.96	120.25
3	95.21	288.87
4	118.21	382.71
5	124.62	406.63

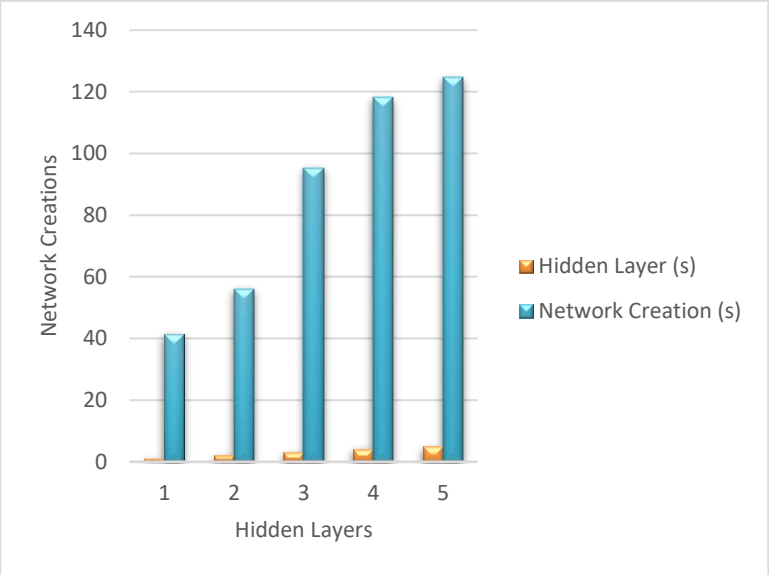

Figure 3. Comparison of different Hidden Layers with Network creations.

The comparison of network creations using different Hidden layers is shown in Table 1 and Figure 3 above. Due to its high network creation in these hidden layers, the suggested algorithm is quite important.

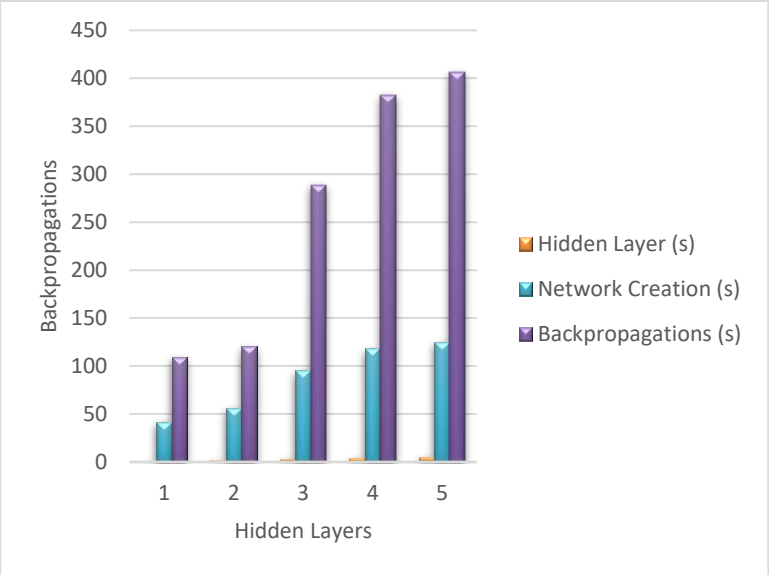

Figure 4. Comparison of different Hidden Layers and Backpropagation.

Table 2. Comparison of different Hidden Layers with Feedforwards and Interactions, Nosie reductions [33].

Hidden Layer (s)	Feedforward (s)	Interactions	Noise Reduction (s)
1	59.80	77	149.03
2	202.12	192	312.27
3	412.85	452	682.82
4	548.42	600	912.07
5	582.13	640	969.58

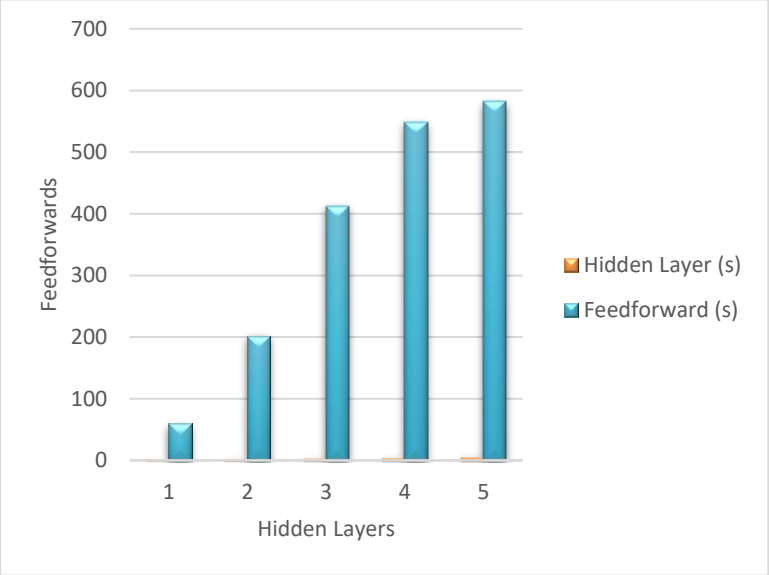

Figure 5. Comparison of different Hidden Layers with Feedforwards.

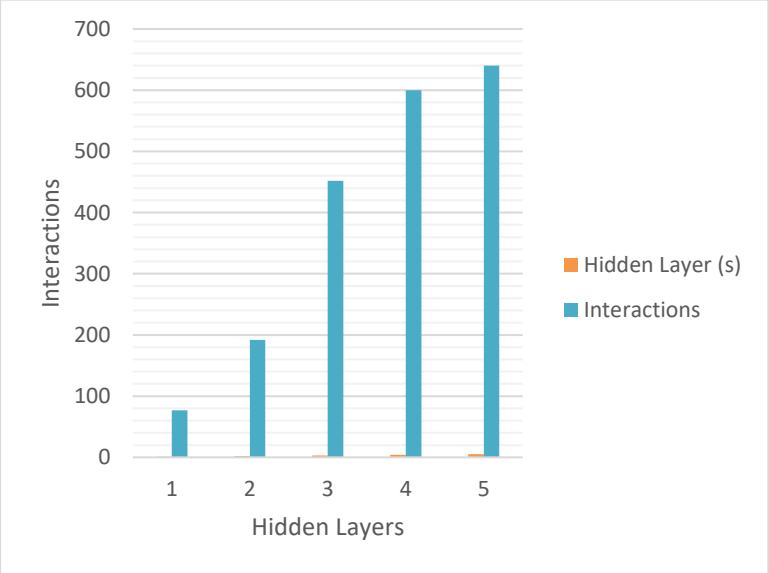

Figure 6. Comparison of different Hidden Layers and Interactions.

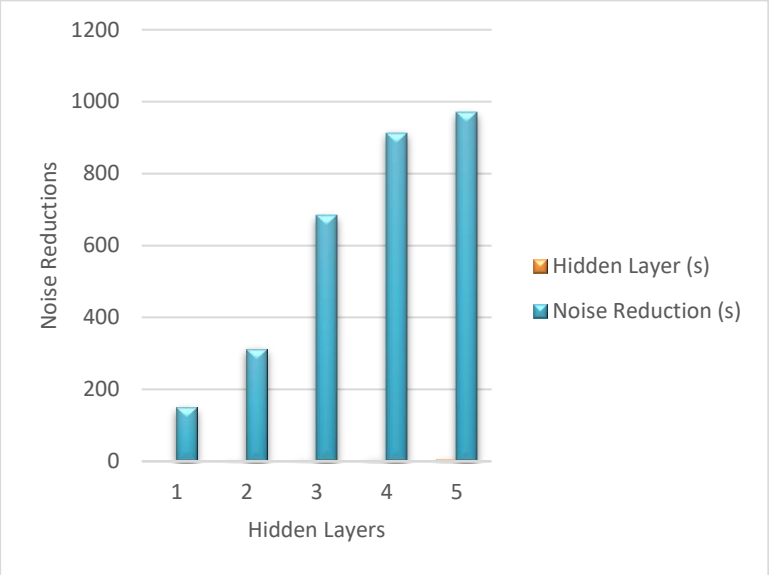

Figure 7. Comparison of different Hidden Layers with Nosie reductions.

The comparison of Feedforward, Interaction, and Noise reduction using different Hidden layers has been shown in Table 2 and Figures 5, 6, & 7 above. Due to its high Feedforward, Interaction, and Noise reduction in these hidden layers, the suggested algorithm is quite important.

Table 3. Comparison of different Hidden Layers with Noise reductions and total time [34].

Hidden Layer (s)	Interaction	Noise Reduction (s)	Total Running Time (s)
1	76	139.02	117.06
2	193	212.17	297.28
3	451	582.72	708.84
4	599	812.06	1045.4
5	639	869.47	1011.60

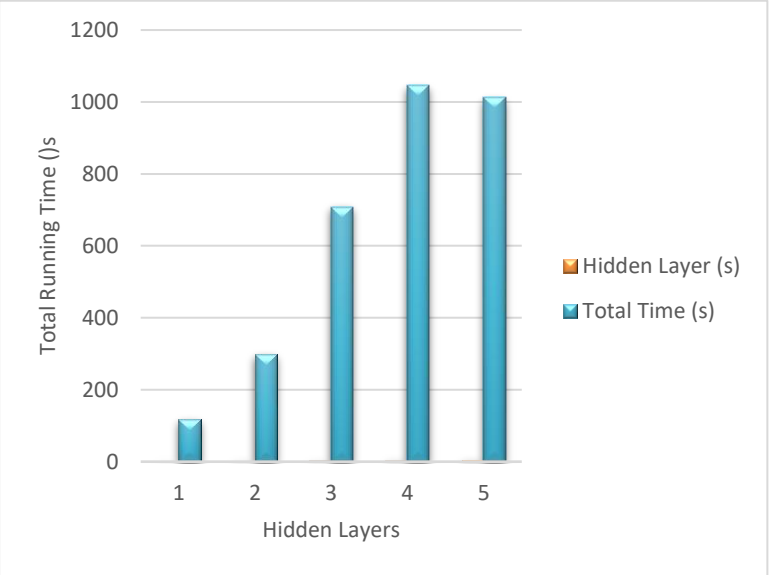

Figure 8. Comparison of different Hidden Layers with total time

The comparison of total running time using different Hidden layers is shown in Table 3 and Figure 8 above. Due to its fast-running time in these hidden layers, the suggested proposed algorithm is quite important.

Table 4. Comparison of different Data sizes with Epoch, Accuracy, and Attack success rate [35].

Data Size	Epoch	Accuracy	Attack success rate
10000	20	88.46 (%)	10.76 (%)
20000	20	91.42	11.26
30000	20	92.81	11.43
10000	35	93.54	12.05
20000	35	94.06	10.83
30000	35	95.77	11.82
10000	60	96.45	12.18
20000	60	97.11	13.63
30000	60	99.05	12.72

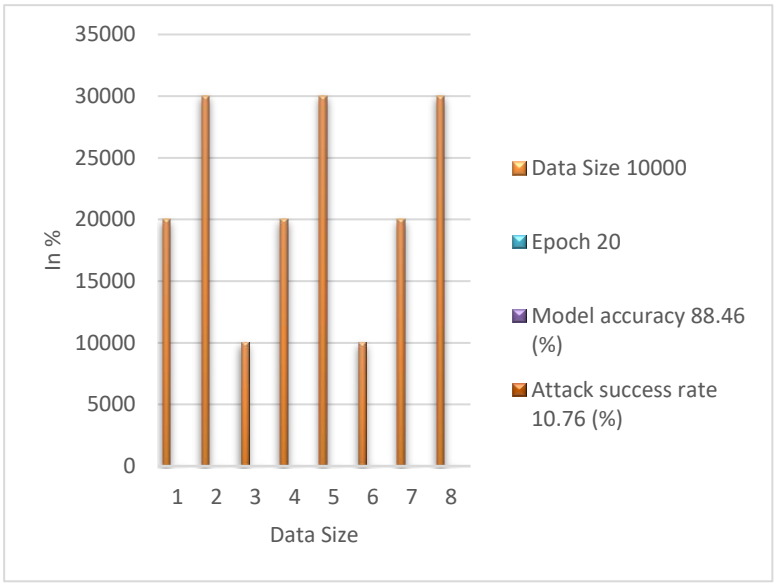

Figure 9. Comparison of different Data sizes with Epoch, Model Accuracy, and Attack success rate.

The comparison of Epoch, Model Accuracy, and Attack success rate using different Data sizes is shown in Table 4 and Figure 9 above. Due to its high Model Accuracy in these data sizes, the suggested proposed algorithm is quite important.

Table 5. Comparison results for Different Datasets with different existing algorithms and proposed algorithms' Accuracy with FPR

Datasets	CNN [33]		ANN [34]		SVM [35]		PSO [36]		Proposed Gaussian-based Neural Network	
	Accuracy	FPR	Accuracy	FPR	Accuracy	FPR	Accuracy	FPR	Accuracy	FPR
Fundoscopy, and	0.75	0.896	0.81	0.874	0.84	0.921	0.87	0.916	0.89	0.884
Chest X-Ray	0.78	0.878	0.83	0.874	0.91	0.959	0.92	0.973	0.94	0.935
Dermoscopy	0.82	0.953	0.86	0.941	0.87	0.956	0.92	0.984	0.96	0.980

The different datasets comparison for the various algorithms, including CNN, ANN, SVM, PSO, and the proposed Gaussian-based Neural Network, is shown in Table 5 above. Due to the rapid runtime in these algorithms, the suggested Gaussian-based Neural Network algorithm is vital.

Table 6. Comparison results for Different Datasets with Accuracy, FPR, and TPR

Data Size	Accuracy(%)	FPR(%)	TPR(%)
10000	88.46	81.27	85.76
20000	91.42	89.30	90.14
30000	92.81	90.19	91.52
10000	93.54	91.25	92.21
20000	94.06	92.89	93.34
30000	95.77	93.51	94.46
10000	96.45	94.68	95.51
20000	97.11	94.89	96.12
30000	99.05	95.47	98.19

The comparison of different datasets for the various metrics, including accuracy, FPR, and TPR, is shown in Table 6 above. Due to its rapid runtime in these metrics, the suggested is vital for its performance. Neural networks have a one-time, highly computational training step. The two primary methods of training are back-propagation and feed-forward. The network performs feed-forward and then back-propagation operations on encrypted instances in order of arrival. After processing each instance, the network generates a model.

Using encrypted data, we use neural networks with varying numbers of layers that are hidden (1, 2, 3, 4, and 5). It implements neural networks in HELib. Along with training the neural network using various batch sizes (282, 576, 1420, 3668, and 6144), it also computes the observed running time increases by 30 times when the batch size is increased. Empirical findings demonstrate that when continuous learning is applied, training over encrypted data is effective, and network performance is tolerable. For instance, with an entire batch of 576 with two hidden layers, we are able to achieve a training pace of 0.68 seconds per instance. For the neural network model with two hidden layers, the training rate drops to 0.10 seconds per instance when the batch size is increased to 6144. It's important to note that while bigger batch sizes speed up training, they also make the network bigger. As a result, we must balance memory usage with execution time, and we should select the appropriate batch size depending on the size within the dataset.

5. CONCLUSION

It is impossible to separate large-scale data training from machine learning performance improvement. It facilitates our daily lives but also increases the likelihood that private information that is sensitive will be revealed. Machine learning privacy is protected by the differential privacy algorithm. Accumulation noise to machine learning that enhances its privacy protection capability is a straightforward way to reduce the model's classification accuracy. In order to address this issue, the article suggests a way of protecting sensitive information that combines deep learning with differential privacy. This approach achieves the goal of deeply protecting users' sensitive data in training datasets. The technique incorporates the concept of differential privacy and also adds noise data while optimizing the network model's parameters. Experimental results demonstrate that privacy modeling settings can be optimized to balance training data sets' privacy protection and availability by examining the differences between attack results and original data. Although there is a guarantee of categorization accuracy, there may be minimal information leakage.

REFERENCES

[1] D. Wang, J. Zhao, and Y. Wang, "A Survey on Privacy Protection of Blockchain: The Technology and Application," *IEEE Access*, vol. 8, pp. 108766–108781, 2020, doi: 10.1109/ACCESS.2020.2994294.

[2] B. A. Malin, "An Evaluation of the Current State of Genomic Data Privacy Protection Technology and a Roadmap for the Future," *J. Am. Med. Informatics Assoc.*, vol. 12, no. 1, pp. 28–34, Oct. 2004, doi: 10.1197/jamia.M1603.

[3] A. R. Miller and C. Tucker, "Privacy Protection and Technology Diffusion: The Case of Electronic Medical Records," *Manage. Sci.*, vol. 55, no. 7, pp. 1077–1093, Jul. 2009, doi: 10.1287/mnsc.1090.1014.

[4] C. Bettini and D. Riboni, "Privacy protection in pervasive systems: State of the art and technical challenges," *Pervasive Mob. Comput.*, vol. 17, pp. 159–174, Feb. 2015, doi: 10.1016/j.pmcj.2014.09.010.

[5] R. Gupta, S. Tanwar, F. Al-Turjman, P. Italiya, A. Nauman, and S. W. Kim, "Smart Contract Privacy Protection Using AI in Cyber-Physical Systems: Tools, Techniques and Challenges," *IEEE Access*, vol. 8, pp. 24746–24772, 2020, doi: 10.1109/ACCESS.2020.2970576.

[6] C. Yin, J. Xi, R. Sun, and J. Wang, "Location Privacy Protection Based on Differential Privacy Strategy for Big Data in Industrial Internet of Things," *IEEE Trans. Ind. Informatics*, vol. 14, no. 8, pp. 3628–3636, Aug. 2018, doi: 10.1109/TII.2017.2773646.

[7] B. Claerhout and G. J. E. DeMoor, "Privacy protection for clinical and genomic data," *Int. J. Med. Inform.*, vol. 74, no. 2–4, pp. 257–265, Mar. 2005, doi: 10.1016/j.ijmedinf.2004.03.008.

[8] Q. Feng, D. He, S. Zeadally, M. K. Khan, and N. Kumar, "A survey on privacy protection in blockchain

system," *J. Netw. Comput. Appl.*, vol. 126, pp. 45–58, Jan. 2019, doi: 10.1016/j.jnca.2018.10.020.

[9] H.-T. Wu and C.-W. Tsai, "Toward Blockchains for Health-Care Systems: Applying the Bilinear Pairing Technology to Ensure Privacy Protection and Accuracy in Data Sharing," *IEEE Consum. Electron. Mag.*, vol. 7, no. 4, pp. 65–71, Jul. 2018, doi: 10.1109/MCE.2018.2816306.

[10] C. Yin, L. Shi, R. Sun, and J. Wang, "Improved collaborative filtering recommendation algorithm based on differential privacy protection," *J. Supercomput.*, vol. 76, no. 7, pp. 5161–5174, Jul. 2020, doi: 10.1007/s11227-019-02751-7.

[11] P. C. Mahawaga Arachchige, P. Bertok, I. Khalil, D. Liu, S. Camtepe, and M. Atiquzzaman, "Local Differential Privacy for Deep Learning," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5827–5842, Jul. 2020, doi: 10.1109/JIOT.2019.2952146.

[12] I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani, "Crowdroid: Behavior-based malware detection system for android," in *Proceedings of the ACM Conference on Computer and Communications Security*, New York, NY, USA: ACM, Oct. 2011, pp. 15–25. doi: 10.1145/2046614.2046619.

[13] A. Abdellatif, M. Wessel, I. Steinmacher, M. A. Gerosa, and E. Shihab, "BotHunter:an approach to detect software bots in GitHub," in *Proceedings of the 19th International Conference on Mining Software Repositories*, New York, NY, USA: ACM, May 2022, pp. 6–17. doi: 10.1145/3524842.3527959.

[14] K. Owusu-Agyemeng, Z. Qin, H. Xiong, Y. Liu, T. Zhuang, and Z. Qin, "MSDP: multi-scheme privacy-preserving deep learning via differential privacy," *Pers. Ubiquitous Comput.*, vol. 27, no. 2, pp. 221–233, Apr. 2023, doi: 10.1007/s00779-021-01545-0.

[15] N. Rajesh and A. A. L. Selvakumar, "Association rules and deep learning for cryptographic algorithm in privacy preserving data mining," *Cluster Comput.*, vol. 22, no. S1, pp. 119–131, Jan. 2019, doi: 10.1007/s10586-018-1827-6.

[16] R. Natarajan, G. H. Lokesh, F. Flammini, A. Premkumar, V. K. Venkatesan, and S. K. Gupta, "A Novel Framework on Security and Energy Enhancement Based on Internet of Medical Things for Healthcare 5.0," *Infrastructures*, vol. 8, no. 2, p. 22, Feb. 2023, doi: 10.3390/infrastructures8020022.

[17] K. I. Jones and S. R, "Information Security: A Coordinated Strategy to Guarantee Data Security in Cloud Computing," *Int. J. Data Informatics Intell. Comput.*, vol. 2, no. 1, pp. 11–31, Mar. 2023, doi: 10.59461/ijdiic.v2i1.34.

[18] Prabhdeep Singh and Ashish Kumar Pandey, "A Review on Cloud Data Security Challenges and existing Countermeasures in Cloud Computing," *Int. J. Data Informatics Intell. Comput.*, vol. 1, no. 2, pp. 23–33, Dec. 2022, doi: 10.59461/ijdiic.v1i2.33.

[19] N. Rajesh and A. A. L. Selvakumar, "Hiding personalised anonymity of attributes using privacy preserving data mining," *Int. J. Adv. Intell. Paradig.*, vol. 7, no. 3/4, p. 394, 2015, doi: 10.1504/IJAIP.2015.073717.

[20] W. Wang, L. Ying, and J. Zhang, "On the Relation Between Identifiability, Differential Privacy, and Mutual-Information Privacy," *IEEE Trans. Inf. Theory*, vol. 62, no. 9, pp. 5018–5029, Sep. 2016, doi: 10.1109/TIT.2016.2584610.

[21] J. Pei, K. Zhong, M. A. Jan, and J. Li, "RETRACTED: Personalized federated learning framework for network traffic anomaly detection," *Comput. Networks*, vol. 209, p. 108906, May 2022, doi: 10.1016/j.comnet.2022.108906.

[22] G. K. Ragesh and A. Kumar, "Trust-based secure routing and message delivery protocol for signal processing attacks in IoT applications," *J. Supercomput.*, vol. 79, no. 3, pp. 2882–2909, Feb. 2023, doi: 10.1007/s11227-022-04766-z.

[23] A. Al Hayajneh, M. Z. A. Bhuiyan, and I. McAndrew, "Improving Internet of Things (IoT) Security with Software-Defined Networking (SDN)," *Computers*, vol. 9, no. 1, p. 8, Feb. 2020, doi: 10.3390/computers9010008.

[24] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *J. Netw. Comput. Appl.*, vol. 84, pp. 25–37, Apr. 2017, doi: 10.1016/j.jnca.2017.02.009.

[25] J. Qiu, Z. Tian, C. Du, Q. Zuo, S. Su, and B. Fang, "A Survey on Access Control in the Age of Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 4682–4696, Jun. 2020, doi: 10.1109/JIOT.2020.2969326.

[26] Dr. S. Smys, Dr. Abul Basar, and Dr. Haoxiang Wang, "Hybrid Intrusion Detection System for Internet of Things (IoT)," *J. ISMAC*, vol. 2, no. 4, pp. 190–199, Sep. 2020, doi: 10.36548/jismac.2020.4.002.

[27] T. Saba, A. Rehman, T. Sadad, H. Kolivand, and S. A. Bahaj, "Anomaly-based intrusion detection system for IoT networks through deep learning model," *Comput. Electr. Eng.*, vol. 99, p. 107810, Apr. 2022, doi: 10.1016/j.compeleceng.2022.107810.

[28] A. Fatani, A. Dahou, M. A. A. Al-qaness, S. Lu, and M. A. Abd Elaziz, "Advanced Feature Extraction and Selection Approach Using Deep Learning and Aquila Optimizer for IoT Intrusion Detection System," *Sensors*, vol. 22, no. 1, p. 140, Dec. 2021, doi: 10.3390/s22010140.

[29] P. Khadivi, T. D. Todd, S. Samavi, H. Saidi, and D. Zhao, "Mobile ad hoc relaying for upward vertical handoff in hybrid WLAN/cellular systems," *Ad Hoc Networks*, vol. 6, no. 2, pp. 307–324, Apr. 2008, doi: 10.1016/j.adhoc.2007.01.005.

[30] E. M. Rudd, A. Rozsa, M. Gunther, and T. E. Boult, "A Survey of Stealth Malware Attacks, Mitigation Measures, and Steps Toward Autonomous Open World Solutions," *IEEE Commun. Surv. Tutorials*, vol. 19, no. 2, pp. 1145–1172, 2017, doi: 10.1109/COMST.2016.2636078.

[31] K. N. Khasawneh, M. Ozsoy, C. Donovick, N. Abu-Ghazaleh, and D. Ponomarev, "Ensemble Learning for Low-Level Hardware-Supported Malware Detection," 2015, pp. 3–25. doi: 10.1007/978-3-319-26362-5_1.

[32] M. B. Bahador, M. Abadi, and A. Tajoddin, "HPCMalHunter: Behavioral malware detection using hardware

performance counters and singular value decomposition," in *2014 4th International Conference on Computer and Knowledge Engineering (ICCKE)*, IEEE, Oct. 2014, pp. 703–708. doi: 10.1109/ICCKE.2014.6993402.

[33] H. Tamura, M. Uchida, M. Tsuru, J. Shimada, T. Ikenaga, and Y. Oie, "Routing Metric Based on Node Degree for Load-Balancing in Large-Scale Networks," in *2011 IEEE/IPSJ International Symposium on Applications and the Internet*, IEEE, Jul. 2011, pp. 519–523. doi: 10.1109/SAINT.2011.96.

[34] R. Elnaggar, K. Chakrabarty, and M. B. Tahoori, "Run-time hardware trojan detection using performance counters," in *2017 IEEE International Test Conference (ITC)*, IEEE, Oct. 2017, pp. 1–10. doi: 10.1109/TEST.2017.8242063.

[35] M. Ozsoy, K. N. Khasawneh, C. Donovick, I. Gorelik, N. Abu-Ghazaleh, and D. Ponomarev, "Hardware-Based Malware Detection Using Low-Level Architectural Features," *IEEE Trans. Comput.*, vol. 65, no. 11, pp. 3332–3344, Nov. 2016, doi: 10.1109/TC.2016.2540634.

[36] B. Zhou, A. Gupta, R. Jahanshahi, M. Egele, and A. Joshi, "Hardware Performance Counters Can Detect Malware," in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, New York, NY, USA: ACM, May 2018, pp. 457–468. doi: 10.1145/3196494.3196515.

BIOGRAPHIES OF AUTHORS

Thanh Chi Phan received his B.S. and M.S. degrees in Computer Science from Hue University, VietNam in 2002 and 2006. He received Ph.D. degree in Technical Pedagogy specialized in Information Technology from Hanoi University of Science and Technology, Vietnam, in 2020. He is a senior lecturer at Quang Tri Teacher Training College and a researcher at Hanoi University of Science and Technology, Vietnam. Fields of interest: His field of research concerns theoretical research on teaching theory and pedagogy of information technology. Application of knowledge science to improve the innovation of technical teaching at universities of technology; application in informatics based on foundation of information and communications technology, Cloud computing in education. Specialization: Teaching Theory and Methodology in Information Technology, Engineering - Information Technology; Teaching method Informatics, Mathematical Theory guarantees to believe study, Analysis, and evaluation in education. He has researched and published some specialized scientific articles published in national journals, and international journals under the category Scopus and ISI. He can be contacted at email: thanhpc.sp@gmail.com

Hung Chi Tran received Computer Science Master's graduate in 2013 with a specialization in IT Engineering. This individual conducted research and oversaw the management of a company's computer network system. Their research focused on data exploration, system security, and the implementation of AI in education system management and operation. He can be contacted at email: hung_tc@qtttc.edu.vn