Computer Modeling in Engineering & Sciences

Tech Science Press

https://doi.org/10.32604/cmes.2026.077784

ARTICLE

# Privacy-Aware Anomaly Detection in Encrypted Network Traffic via Adaptive Homomorphic Encryption

**Yu-Ran Jeon**[1], **Seung-Ha Jee**[1], **Su-Kyoung Kim**[1] **and Il-Gu Lee**[1,2,*]

[1]Department of Future Convergence Technology Engineering, Sungshin Women's University, Seoul, Republic of Korea
[2]Department of Convergence Security Engineering, Sungshin Women's University, Seoul, Republic of Korea

*Corresponding Author: Il-Gu Lee. Email: iglee@sungshin.ac.kr

Received: 16 December 2025; Accepted: 23 February 2026; Published: 30 March 2026

**ABSTRACT:** As cyberattacks become increasingly sophisticated and intelligent, demand for machine-learning-based anomaly detection systems is growing. However, conventional systems generally assume a trusted server environment, where traffic data is collected and analyzed in plaintext. This assumption introduces inherent privacy risks, as privacy-sensitive information may be exposed if the server is compromised or misused. To address this limitation, privacy-preserving anomaly detection approaches have been actively studied, enabling anomaly detection to be performed directly on encrypted traffic without revealing privacy-sensitive data. While these approaches offer strong confidentiality guarantees, they suffer from significant drawbacks, including substantial computational overhead, high latency, and degraded detection accuracy. To overcome these limitations, we propose a privacy-aware anomaly detection (PAAD) model that adaptively applies homomorphic encryption based on the privacy sensitivity of incoming traffic. Instead of encrypting all data indiscriminately, PAAD dynamically determines whether traffic should be processed in plaintext or ciphertext and performs homomorphic inference only for privacy-sensitive data. This selective encryption strategy effectively balances privacy protection and system efficiency. Extensive experiments conducted under diverse network environments demonstrate that the proposed PAAD model significantly outperforms conventional anomaly detection models. In particular, PAAD improves detection accuracy by up to 73%, reduces latency by up to 8.6 times, and achieves negligible information leakage, highlighting its practicality for real-world privacy-sensitive network monitoring scenarios.

**KEYWORDS:** Homomorphic encryption; machine learning; privacy-aware anomaly detection

## 1 Introduction

With recent advancements in digital technology, cyberattack techniques have become increasingly sophisticated, which highlights the importance of anomaly detection systems for identifying these attacks [1,2]. Machine-learning-based anomaly detection system learns normal patterns from given data and then detects values that deviate significantly from the data distribution, thereby identifying abnormal or potentially dangerous events [3]. This technology has been applied across various domains, such as the financial sector for detecting abnormal transactions [4,5] and the corporate sector for detecting unauthorized internal access [6,7].

However, machine-learning-based anomaly detection systems have been designed based on the assumption that servers are trustworthy, with data collected in plaintext on servers and then used in the learning and detection processes. Although this structure is simple to implement, it has a fundamental

limitation in that data is exposed to the server [8]. In particular, privacy-sensitive information, such as financial transaction histories, medical records, and location information, can be leaked by attackers or misused in cyberattacks. Therefore, with the zero-trust security paradigm gaining attention, privacy-preserving anomaly detection systems have been actively studied [9]. Homomorphic encryption (HE) is a cryptographic technique that allows arithmetic operations to be performed directly on ciphertext, ensuring that the decrypted result is identical to the result obtained by applying the same operation to plaintext [10]. Anomaly detection with HE enables privacy-preserving learning in encrypted form without exposing sensitive information to the server as plaintext [11]. However, operating over encrypted data involves a significantly higher processing burden than general operations, which increases latency and reduces detection accuracy because of the noise generated during the complex operation process [12]. Hence, although HE preserves privacy, it introduces considerable computational overhead in anomaly detection scenarios, which require real-time performance [13].

To address this tradeoff, this study proposes the privacy-aware anomaly detection (PAAD) model. This anomaly detection method dynamically applies HE based on the privacy sensitivity of data. PAAD balances privacy and performance by performing homomorphic operations only when necessary, considering traffic sensitivity.

The main contributions of this study are as follows:

- An anomaly detection system that adaptively performs homomorphic operations is proposed to address the privacy limitations of conventional plaintext-based anomaly detection systems and the latency and accuracy issues of privacy-preserving anomaly detection systems.
- The tradeoff between latency and privacy is improved by classifying traffic according to its privacy sensitivity and selecting anomaly detection methods based on traffic type.
- Compared to conventional anomaly detection models, we demonstrate efficient anomaly detection by improving accuracy by up to 73%, reducing delay by a factor of 8.6, and achieving low information leakage.

The remainder of this paper is organized as follows. Section 2 analyzes research on machine-learning-based and privacy-preserving anomaly detection, and Section 3 describes the privacy threat model. Section 4 proposes a technique that dynamically applies HE based on the privacy sensitivity of the data. Section 5 presents an analysis of the performance evaluation results of the proposed and conventional approaches. Finally, Section 6 concludes the study.

## 2 Related Work

This section analyzes previous studies on machine-learning-based anomaly detection and privacy-preserving anomaly detection technologies, and federated learning-based privacy protection techniques, as listed in Table 1. Machine-learning-based anomaly detection receives plaintext as input and performs learning and inference, whereas privacy-preserving anomaly detection receives homomorphically encrypted data as input and performs anomaly detection using HE.

**Table 1:** Previous studies on anomaly detection methods.

| Category | Ref. | Contributions | Limitations | Encryption Status | Dynamic Encryption Status |
|---|---|---|---|---|---|
| Machine-learning-based anomaly detection | [14] | • Two fuzzy anomaly detection techniques using unsupervised and supervised learning were proposed to address the boundary uncertainty problem.<br>• These techniques are more efficient with lower computational complexity than one-class support vector machine (OC SVM) and support vector data description (SVDD) models. | • Learning process relied on plaintext data; hence, data security was not taken into consideration. | X | X |
| | [15] | • An anomaly detection technique based on system logs for virtual network functions was proposed.<br>• It can predict abnormal signs up to 35 min before the actual failure caused by an attack occurs. | • Normal logs with unexpected pattern variations may be falsely detected as anomalies. | X | X |
| | [16] | • An ensemble-learning-based anomaly detection technique was proposed to combine the strengths of multiple detection models and improve detection accuracy in cloud environments. | • The integration of multiple detection models increases computational complexity, which leads to higher latency. | X | X |
| | [17] | • A network anomaly detection technique using a one-dimensional convolutional neural network (1D CNN) and the synthetic minority over-sampling technique (SMOTE) was proposed to address the class imbalance problem in network traffic data.<br>• The model achieved a higher F1 score than existing CNN-based methods. | • SMOTE may generate overly similar samples, resulting in overfitting and reducing generalization to new attack types.<br>• Protocol-wise training increases computational cost. | X | X |
| | [18] | • A federated anomaly detection technique using noisy global density estimation and self-supervised ensemble distillation was proposed.<br>• It improves detection accuracy while preserving privacy through noise-added density sharing. | • Multi-stage training and frequent communication cause high computational overhead. | X | X |
| | [19] | • A polynomial approximation technique optimized for HE was proposed by composing multiple low-degree polynomials.<br>• It demonstrated performance similar to the conventional model without separate retraining. | • Privacy-preserving machine learning increases latency and is difficult to apply in real environments. | O | X |

(Continued)

**Table 1 (continued)**

| Category | Ref. | Contributions | Limitations | Encryption Status | Dynamic Encryption Status |
|---|---|---|---|---|---|
| Privacy-preserving anomaly detection | [20] | • A real-time intrusion detection system that performs distributed learning on homomorphically encrypted data was proposed.<br>• The latency of the detection process was reduced through distributed learning. | • It reduces latency compared to conventional centralized learning on homomorphically encrypted data, but still has higher latency. | O | X |
| | [21] | • A federated learning technique using discrete random error learning (DREL)-based quantum processes and Brakerski–Fan–Vercauteren (BFV) HE schemes was proposed to prevent information leakage.<br>• It enhances integer operation efficiency and model weight security. | • It was not evaluated, in terms of accuracy, communication overhead, or computational overhead, against lightweight cryptographic schemes that do not use HE. | O | X |
| | [22] | • A HE-based histogram design for anomaly detection was proposed.<br>• The algorithm resolves loop and conditional statement calculations that were previously impossible with the conventional HE scheme. | • Compiling and executing the proposed technique requires several minutes. | O | X |
| | [23] | • Privacy-preserving machine learning (PPML) is proposed by applying the residue number system (RNS)–based Cheon–Kim–Kim–Song (CKKS) HE scheme.<br>• It resolves the noise accumulation issue in HE. | • The proposed model incurs a relatively long execution time. | O | X |
| Privacy protection based on federated learning models | [24] | • Solves the dropout problem occurring in environments where client models exist within mobile devices.<br>• Proposes a protocol combining efficient HPRG with Shamir Secret Sharing techniques while maintaining weight protection for clients. | • Global models at intermediate and final steps must be disclosed to all model participants, making them vulnerable to membership inference attacks. | O | X |
| | [25] | • Proposes Fed-ANIDS, an intrusion detection framework combining Autoencoder and Federated Learning (FL) to address data privacy, detection accuracy, and data quality issues in NIDS.<br>• Enables training a global model through distributed client collaboration without transmitting data to a central server. | • Does not consider the potential leakage of sensitive data during model inference as the number of clients increases.<br>• Does not consider the leakage of weights during transmission to the central server or the associated costs. | O | X |

Refs. [14–18] focus on machine learning–based anomaly detection techniques. Ouyang and Zhang [14] proposed a technique to address the anomaly data detection uncertainty problem by applying fuzzy theory and particle computing in the anomaly data detection process to mitigate dataset imbalance. Detection performance was improved by optimizing the anomaly detection boundary using the IG technique. However, whether the IG technique is the optimal approach for anomaly data detection remains unclear; moreover, the risk of information leakage from using plaintext data during model training has not yet been considered. Rim et al. [15] proposed a technique for detecting abnormal signs before system failure by analyzing text logs generated from virtualized network functions. The method vectorizes log data and merges duplicate entries generated within the same process into a single vector. This enables early prediction of abnormal signs. However, merging similar logs into a single vector increases the error range of the classification criterion, rendering it difficult to distinguish normal and abnormal data in detail. Xin et al. [16] proposed an ensemble-learning-based anomaly detection technique to improve detection accuracy in cloud applications. The technique combines four unsupervised detection methods using linear and deep ensemble structures. The deep ensemble model achieved an F1 score of 0.83 and could detect performance anomalies up to 4 min before they occurred. However, because the approach requires training and integrating multiple base detectors within a neural network, the computational cost increases significantly, resulting in higher detection latency. Hooshmand and Hosahalli [17] proposed a network anomaly detection technique using a 1-D CNN combined with SMOTE to address class imbalance in network traffic data. The model was trained separately for transmission control protocol (TCP), user datagram protocol (UDP), and other traffic types, improving detection accuracy for minority attack classes and enabling it to achieve a higher F1 score than existing CNN-based methods. However, because SMOTE generates synthetic samples from existing data, it may produce overly similar samples, causing overfitting and reducing generalization to new attack types. Dong et al. [18] proposed a federated anomaly detection technique that combines noisy global density estimation with self-supervised ensemble distillation to address false and missing detection issues in FL environments. The technique aligns anomaly definitions among clients by sharing noise-added density functions and aggregates local model capacities, improving detection accuracy while preserving data privacy. However, excessive noise can distort shared density functions and weaken the model's ability to distinguish normal and abnormal samples.

Refs. [19–23] investigate privacy-preserving anomaly detection approaches. Lee et al. [19] proposed a polynomial technique that approximated the ReLU and Max-pooling functions to enable HE-based data learning and address high model latency and bootstrapping errors caused by fully homomorphic encryption (FHE). In conventional methods that utilize homomorphically encrypted data, the network of a predefined model must either be redefined or retrained. In contrast, the proposed technique eliminates the need for retraining or an additional model design. However, the model exhibits significant latency, with a maximum inference time of 4764 s per image, indicating that further optimization is required for real-world deployment of the approximate deep learning model. Duch Manh et al. [20] applied HE to protect data in a blockchain environment and proposed a privacy-preserving distributed learning technique to improve model latency. The technique distributes encrypted data from a central server to multiple worker nodes, trains models independently on each node, and aggregates the results at the central server. Although the model reduces latency, it still exhibits a latency of at least 21.64 h. Castro et al. [21] proposed a privacy-preserving FL strategy based on HE to prevent leakage of shared parameters from local models. The proposed model introduces a DREL process to compute local model weights and encode them as integers. By utilizing the BFV scheme, the technique protects local model weights while enabling effective computation. However, comparative evaluations using lightweight encryption schemes were not conducted. Lazea et al. [22] proposed a technique for constructing equal-width histograms and detecting anomalies in homomorphic encrypted data generated by

Internet of Things (IoT) devices. Their approach employs an algorithm that determines bucket membership without relying on loop- or conditional-based operations. However, when encryption was performed using the proposed technique, the encryption overhead increased by two million times compared to the original data, and it consumes a significant amount of time (approximately 8000 s) for compilation and execution. Lee et al. [23] used a bootstrapping technique to address the noise problem that arises when computations are performed on encrypted data in FHE-based PPML. Although HE enables computation on ciphertext, noise accumulates as operations progress. Bootstrapping resets the noise, enabling deeper computations. However, the proposed technique requires approximately 3 h to infer a single image, indicating that although accuracy improves, latency remains significantly higher than plaintext-based models.

Refs. [24,25] investigates a framework for privacy in FL models. Liu et al. [24] propose a protocol combining a Homomorphic Pseudorandom Generator (HPRG) and Shamir Secret Sharing techniques to address communication, computational costs, and the Dropout problem while preserving privacy in large-scale FL environments involving mobile and edge devices. By using an HPRG-based masking scheme instead of the traditional Diffie-Hellman key exchange and incorporating Shamir Secret Sharing, they securely manage clients' weights while improving the computational speed of the global model. Even if a dropout occurs among clients, only the sum of masks from connected clients is computed. Consequently, even with a dropout rate exceeding 10% among 500 clients, execution speeds were up to 6.37 times faster than those of conventional methods. However, a limitation exists: the global model at intermediate or final stages must be disclosed to all clients, making it vulnerable to membership inference attacks. Idrissi et al. [25] propose FL for anomaly-based network intrusion detection systems (Fed-ANIDS) to address data privacy, detection accuracy, and model training data quality issues in Network Intrusion Detection Systems (NIDS). Fed-ANIDS utilizes an Autoencoder to classify normal and attack traffic on the network based on Reconstruction Error. It employs the Federated Proximal (FedProx) algorithm to enhance model training stability even under imbalanced training data conditions across clients. The proposed technique demonstrated performance improvements of up to 37.32% in F1-Score compared to detection models based on Generative Adversarial Networks (GAN) and Bidirectional GAN (BiGAN). However, as the number of clients increases, the attack surface for adversaries to perform model inference attacks for sensitive data leakage also expands. The proposed method does not address this security concern. Furthermore, it does not account for the cost of transmitting weights among clients.

An analysis of prior studies reveals two fundamental limitations in existing anomaly detection approaches. On one hand, machine-learning-based anomaly detection methods pose a risk of privacy leakage by exposing sensitive data during detection. On the other hand, privacy-preserving anomaly detection methods with HE effectively protect sensitive information but incur substantial computational complexity, as HE is typically applied uniformly to all traffic regardless of its privacy sensitivity. Consequently, data that requires privacy protection must not be exposed in plaintext during anomaly detection, and encrypted inference becomes essential for such data. To address these limitations, this study proposes an anomaly detection technique that dynamically applies HE by selectively performing encrypted or plaintext inference based on the privacy sensitivity of the data.

## 3 Privacy Threat Model

In this section, we describe the privacy threat model regarding how attackers execute attacks and the data leakage paths. Attackers collect both sensitive and non-sensitive traffic, which are inputs to PAAD, through packet sniffing and man-in-the-middle attacks. Additionally, traffic from the Internet of Medical Things (IoMT) or traffic that can be used to infer sensitive personal information, such as video streaming, also becomes a target for attackers [26,27]. Based on the collected traffic, attackers can leak personal information

through membership inference attack scenarios [28]. Fig. 1 illustrates the sensitive data leakage path and attack methods.

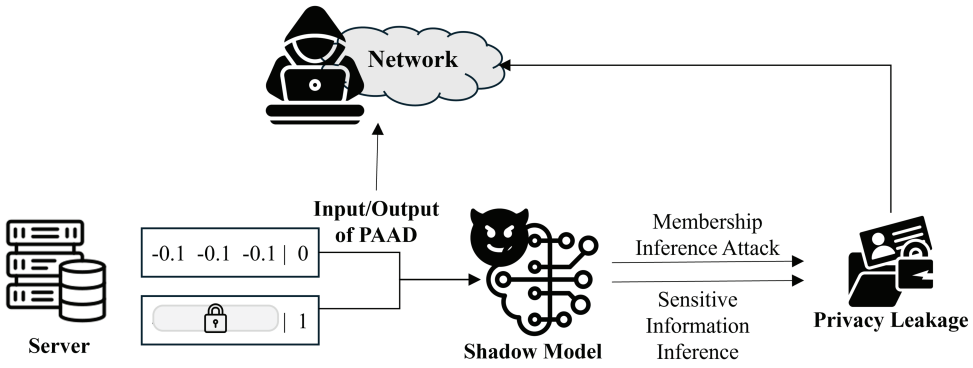


**Figure 1:** Sensitive data leakage path and attack scenario.

The attacker specifically collects traffic sent to the PAAD model to determine whether specific data was included in the model's training dataset. If the collected data is in plaintext form, the attacker can easily infer the user's identity or sensitive medical information. Furthermore, the attacker can eavesdrop on input and output values of the target PAAD model to build a Shadow Model, thereby stealing sensitive information from legitimate users. Since the Shadow Model mimics PAAD, attackers can exploit it by inputting plaintext traffic to reverse-engineer sensitive information, leading to severe privacy violations.

In addition to membership inference attacks, privacy threats such as model inversion attacks may also affect network anomaly detection systems. Model inversion attacks attempt to reconstruct sensitive input features by exploiting access to model inputs or outputs.

In the proposed PAAD framework, such attacks are partially mitigated because privacy-sensitive traffic is processed under HE, preventing attackers from directly observing plaintext inputs or inference results.

## 4 Efficient Privacy-Aware Anomaly Detection

This section describes the operational structure and principles of the proposed PAAD model. When a user accesses a service, traffic is generated through various activities such as searching, browsing, logging in, and transmitting data; some of this traffic contains privacy-sensitive information, including personal or authentication data. The server collects and learns the traffic generated when a user interacts with an application or web service. It then uses real-time traffic as input for anomaly detection. The architecture of the proposed model is shown in Fig. 2.

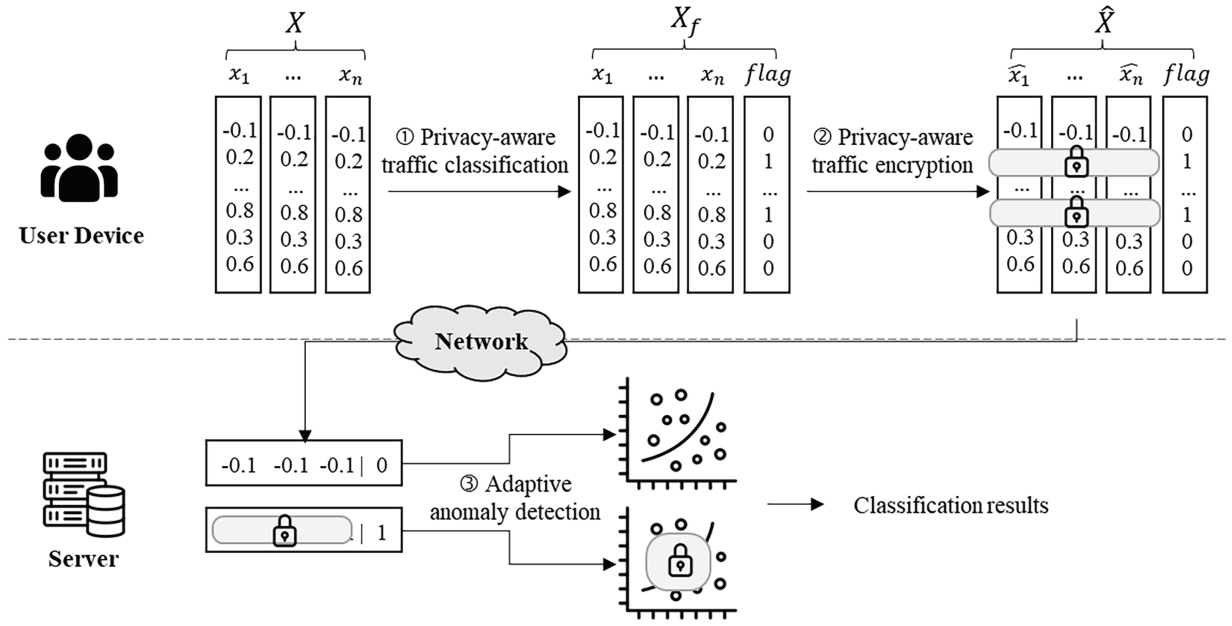


**Figure 2:** Architecture of the proposed PAAD.

### *4.1 Privacy-Aware Traffic Encryption*

When using application services, users perform actions such as account logins and web browsing; the application collects traffic generated from the user's actions in real time. Traffic is classified based on its privacy sensitivity. If the traffic contains private information, it is classified as privacy-sensitive; if it does not contain private information, it is classified as non-sensitive. The PAAD model assigns a flag to each data instance according to its privacy sensitivity. This flag is determined based on predefined rule-based criteria that assess whether the traffic contains personally identifiable information. A value of 1 is assigned to privacy-sensitive data, and a value of 0 is assigned to non-sensitive data. At the user application layer, data instances with a privacy flag of 1 are encrypted before transmission, whereas data instances with a privacy flag of 0 are transmitted to the server in plaintext. The server then performs anomaly detection on the received data according to its encryption status. Features related to user identification, network identifiers, or location information are treated as privacy-sensitive data, whereas features representing general session characteristics are treated as non-sensitive data.

In the proposed framework, encryption decisions are determined at the application layer of each node, prior to data transmission, based on the sensitivity of individual traffic instances. This design choice reflects a practical security assumption in which the server performing machine learning is not fully trusted. Accordingly, the proposed mechanism enables sensitivity-aware adaptive HE while mitigating potential privacy leakage during the sensitivity inference process.

### *4.2 Model Training*

The PAAD model adaptively performs encrypted inference based on the privacy sensitivity of the incoming traffic. Accordingly, the server pretrains two anomaly detection models: an encrypted model for encrypted inference and a plaintext model for plaintext-based inference. The encrypted anomaly detection model was derived from the plaintext-trained model by utilizing its homomorphically encrypted weights

and biases. Using these encrypted parameters, the model enables privacy-preserving inference of encrypted data without requiring decryption.

A logistic regression model was adopted as the learning and classification model for traffic anomaly detection [29]. The model predicts classes in the form of integer values and classifies them based on the probability that the data belong to a given class. This enables the model to estimate the probability of normal or abnormal traffic directly, enabling anomaly detection. In addition, logistic regression is particularly well suited for HE–based inference, as it primarily relies on linear operations and a sigmoid activation function, which can be efficiently approximated using low-degree polynomials. The number of epochs was set to 10 when training the logistic regression model, and the decision threshold for distinguishing between abnormal and normal values during inference was set to 0.5.

### *4.3 Adaptive Anomaly Detection*

The traffic processed by the application according to predefined rules is sent to an anomaly detection server. The server inspects the incoming traffic and determines whether it is privacy-sensitive based on the traffic flag value. A homomorphic operation is used to detect anomalies when privacy-sensitive traffic is received as input. When non-sensitive traffic is received, plaintext operations are used for anomaly detection. For privacy-sensitive traffic, the pretrained encrypted model produces encrypted outputs through homomorphic operations; the final anomaly determination is made after decrypting these results. Algorithm 1 presents the pseudocode of the proposed model.

**Algorithm 1:** Pseudocode of the proposed model

**Input:**
X = $\{x_1, x_2, \ldots, x_n\}$ // incoming traffic data
$flag_{privacy}(x_i) \rightarrow$ {0: non sensitive, 1: privacy-sensitive} // the privacy sensitivity of traffic
**Output:**
y $\in$ $\{0, 1\}$ // classification result
**Step 1**: Data collection
for $x_i$ in X do
    if $flag_{privacy}(x_i)$ == 1 then
        enc_$x_i$ $\leftarrow$ Encrypt($x_i$, HE_context)
        Store(enc_$x_i$)
    else
        Store($x_i$)
    end if
end for
**Step 2**: Model training
model $\leftarrow$ LogisticRegression() // plaintext-based anomaly detection model
enc_model $\leftarrow$ EncryptedLR(model) // anomaly detection model with HE
**Step 3**: Adaptive anomaly detection
for $x_i$ in X do
    if $flag_{privacy}(x_i)$ == 0 then
        pred $\leftarrow$ model.predict($x_i$) // Plaintext inference

(Continued)

**Algorithm 1 (continued)**

```
    else
        pred ← model.predict(enc_x_i) // Encrypted inference
    end if
    if pred > decision_threshold then
        y ← 1 // classification result of x_i is anomaly
    else
        y ← 0 // classification result of x_i is normal
    end if
end for
```

## 5 Evaluation and Analysis

In this section, we compare the performance of the proposed and conventional models in various network environments. The proposed model randomly receives plaintext and ciphertext traffic as input and dynamically detects anomalies based on the type of incoming traffic. For privacy-sensitive traffic, HE using the TenSEAL library was applied to preserve privacy, and anomaly detection was performed through homomorphic operations [30,31]. TenSEAL is a Python library built on Microsoft SEAL that supports efficient machine learning and inference on data encrypted with the CKKS scheme [32]. In our implementation, the sigmoid activation function was approximated using a degree-3 polynomial. The CKKS encryption parameters were configured with a polynomial modulus degree $n = 4096$ and a coefficient modulus composed of three primes with bit sizes $[40, 20, 40]$. The global scaling factor was set to $2^{10}$ to balance numerical precision and computational efficiency. Galois keys are generated to enable encrypted vector operations required for homomorphic dot-product computations. All remaining cryptographic parameters are set to default values provided by the TenSEAL library.

To simulate an authenticated traffic collection environment, we used the WebAuthAnomalyDetection dataset, which consists of synthetic web application authentication traffic labeled as normal and anomalous login events [33]. The anomalous class represents unusual activities, such as repeated failed login attempts, abnormal session durations, or deviations from typical user behavior patterns, which may indicate unauthorized access attempts or system failures. A description of the features of the dataset is provided in Table 2. This dataset comprises synthetic data generated from user authentication logs of web applications and contains traffic related to normal and abnormal login events. It consists of nine features that represent characteristics of login attempts: 'USER ID', 'Timestamp', 'Login Status', 'IP Address', 'Device Type', 'Location', 'Session Duration', 'Failed At-tempts', and 'Behavioral Score'. The features related to user identification, network identifiers, or location information are treated as privacy-sensitive data, whereas features representing general session characteristics are treated as non-sensitive data. In the original dataset [33], the number of anomalous samples is significantly smaller than that of normal samples. Therefore, we applied down-sampling strategy to mitigate the effect of class imbalance and ensure a fair evaluation of anomaly detection performance.

**Table 2:** Features of the dataset.

| Feature | Description |
|---|---|
| User ID | Unique identifier for each user |
| Timestamp | Date and time of the login attempt |
| Login status | Status of success or failure of the login attempt |
| IP address | IP address used for the login attempt |
| Device type | Type of device used for the login attempt |
| Location | Geographical location of the login attempt |
| Session duration | Length of the session in seconds |
| Failed attempts | Number of unsuccessful attempts prior to a successful login |
| Behavioral score | Score that represents the user's normal behavior patterns, used to detect anomalies |

### *5.1 Evaluation Metrics*

To verify the performance of the proposed model across various network environments, we evaluated its accuracy, latency, information leakage, and efficiency based on the data-sampling ratio and the proportion of privacy-sensitive data. Accuracy and latency were measured by determining the inference accuracy and time when new data were input into the trained model.

Information leakage is defined as any instance in which privacy-sensitive traffic is handled in plaintext using an anomaly detection model. Eqs. (1)–(3) present the formulas used to evaluate information leakage in our system. The privacy leakage indicator for each packet is defined in Eq. (1). Let $s_i \in \{0,1\}$ denote whether the $i$-th packet contains privacy-sensitive information, and $e_i \in \{0,1\}$ denote whether HE is applied to the packet. When a privacy-sensitive packet is processed without encryption, the privacy leakage indicator $L_i$ takes the value 1.

$$\text{Privacy leakage indicator } (\boldsymbol{L_i}) = \begin{cases} 1, & \text{if } s_i = 1 \text{ and } e_i = 0 \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Using this indicator, the total number of leaked packets was calculated as the sum of $L_i$ across all $N$ packets.

$$\text{Total number of leaked packets } = \sum_{i=1}^{N} L_i. \tag{2}$$

Finally, the information leakage rate ($ILR$) is calculated as the ratio of the leaked packets to the total number of packets.

$$ILR = \frac{\text{Total number of leaked packets}}{\text{Total number of packets}}. \tag{3}$$

Consequently, as HE is applied more extensively, the amount of information leakage decreases, whereas the more frequent use of plaintext operations leads to greater information leakage.

The efficiency of the model ($M_{eff}$) was evaluated by comprehensively considering the anomaly detection accuracy ($Acc$), latency ($d$), information leakage rate ($ILR$), and privacy-sensitive packet processing

rate ($SPR$). As these metrics have different scales, each value was normalized to the range $[0,1]$. We denote normalized metrics with an overbar. The efficiency of the model is defined as

$$M_{eff} = \overline{Acc} \cdot (1 - \overline{d}) \cdot (1 - \overline{ILR}) \cdot \overline{SPR}. \tag{4}$$

Eq. (4) indicates that a model is considered more efficient when it achieves a higher detection accuracy and privacy-sensitive packet processing rate while incurring lower latency and lower information leakage. In this study, the efficiency of models is evaluated and compared fairly by assigning equal weights to $Acc$, $d$, $ILR$, and $SPR$. This weighting strategy is adopted to avoid bias toward any single metric and to reflect a balanced trade-off among detection performance, computational efficiency, and privacy preservation. Accordingly, the proposed efficiency metric is flexible and can be adapted to application-specific requirements by adjusting the weight assigned to each metric. For example, in privacy-sensitive environments such as financial or healthcare systems, higher weights can be assigned to the information leakage to prioritize privacy protection. In contrast, in scenarios where real-time intrusion detection is critical, the latency metric can be emphasized to optimize the model for timely response.

### *5.2 Comparative Experiments*

In this section, we describe the simulations of environments with varying amounts of traffic data and different proportions of privacy-sensitive data. Plaintext-based anomaly detection (PBAD) is a conventional model that performs anomaly detection through plaintext-based inference, whereas homomorphic encrypted anomaly detection (HEAD) is a conventional model that performs encrypted inference to privately preserve anomaly detection.

Fig. 3 illustrates the accuracy and latency of the comparative models as a function of the data-sampling ratio. The proportion of privacy-sensitive data in the dataset was set to 0.5. As the data sampling ratio increases, the amount of training data increases accordingly, which improves the accuracy of the PBAD and PAAD models. However, the HEAD model maintains low accuracy regardless of the data volume. The PAAD model exhibits lower accuracy than the PBAD model; it improves the accuracy by 42% compared to the HEAD model. In terms of the latency, the HEAD model exhibits the highest delay, followed by the PAAD and PBAD models. This is because the HEAD model always performs HE-based anomaly detection regardless of traffic type, which introduces substantial computational overhead and significant latency. In environments with excessive traffic, the HEAD model incurs 90.8% higher latency than the PAAD model. These results show that the proposed PAAD model significantly improves accuracy and latency compared to the HEAD model by adaptively performing homomorphic operations. As higher data sampling ratios represent large-scale network environments with increased traffic volumes, these results show that PAAD can efficiently scale to handle such scenarios.

Fig. 4 presents the results of evaluating accuracy and latency as the proportion of privacy-sensitive data increases. We evaluated the proposed PAAD model under different misclassification rates (mcr), specifically mcr = 0.1 and mcr = 0.3, to analyze the impact of privacy sensitivity misclassification. To isolate the effect of HE usage from the privacy-sensitive data ratio, we introduce a random homomorphic encryption-based anomaly detection (RHAD), which applies HE to a randomly selected subset of samples without considering data sensitivity. For instance, when the privacy-sensitive data ratio is 10%, RHAD applies HE to 10% of the randomly selected samples. Conventional models process traffic without considering privacy sensitivity; consequently, the PBAD model achieves approximately 92% accuracy, whereas the HEAD model achieves around 50%. As RHAD and PAAD apply HE to a similar proportion of data, they exhibit comparable accuracy levels, both achieving approximately 40% higher accuracy than the HEAD model. When the

privacy-sensitive data ratio is below 50%, lower misclassification rates enhance accuracy and reduce latency by enabling more precise selective encryption. Conversely, when privacy-sensitive traffic dominates, higher misclassification reduces the number of homomorphic operations, improving performance at the cost of increased privacy risk. Overall, these results reveal a clear trade-off between system performance and privacy protection under imperfect sensitivity classification. Regarding latency, the HEAD model exhibits the highest inference delay, followed by the PAAD and PBAD models. PAAD achieves a latency between those of HEAD and PBAD by accounting for data sensitivity and applying HE only to sensitive data.

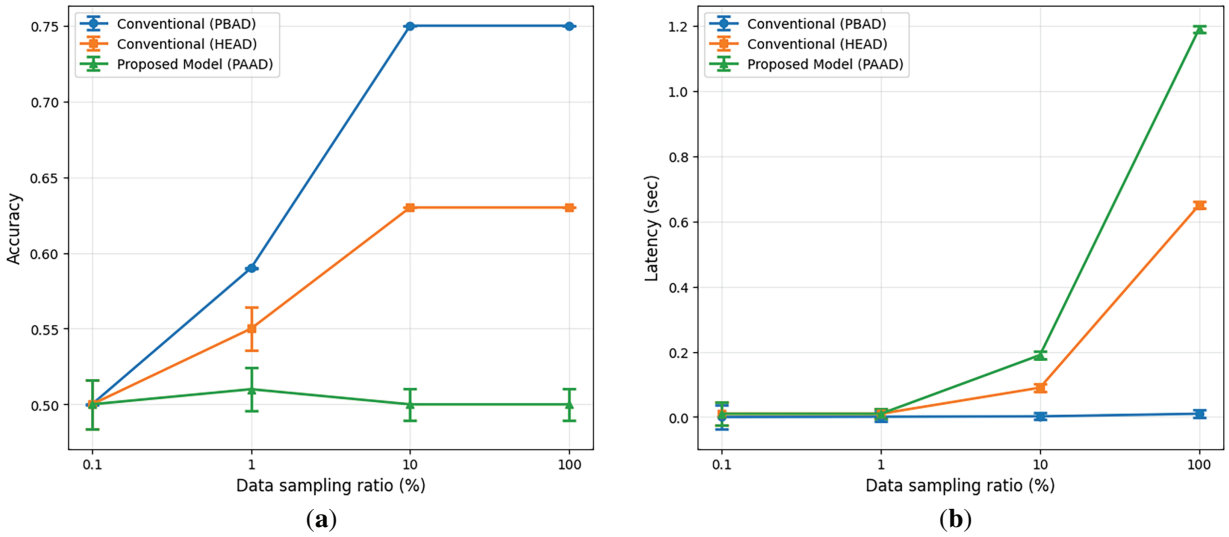


**Figure 3:** (**a**) Accuracy and (**b**) latency vs. data sampling ratio of the comparative models used in the study.

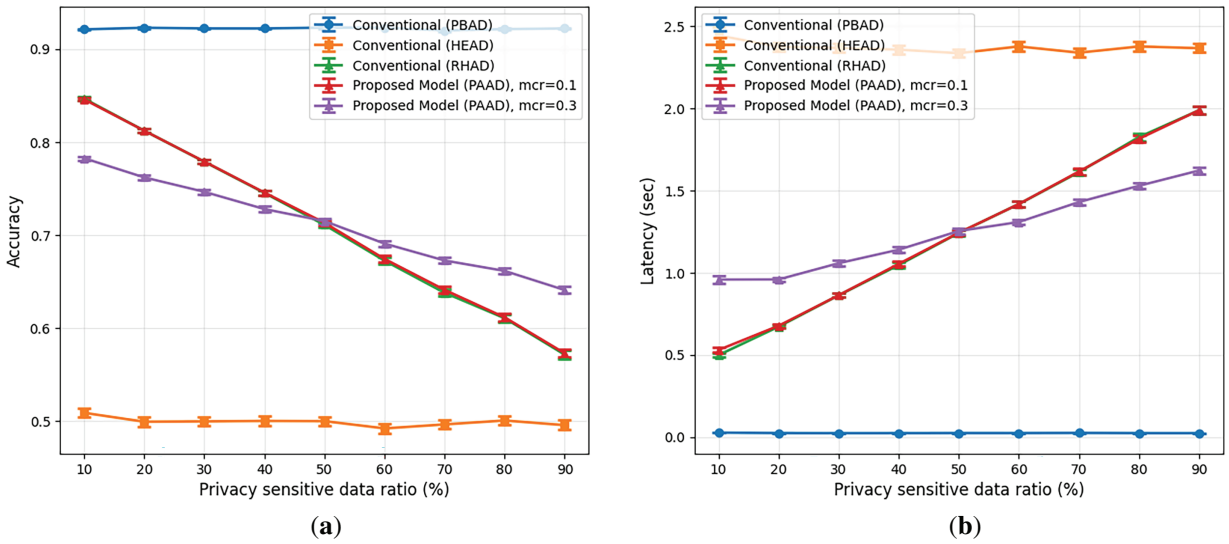


**Figure 4:** (**a**) Accuracy and (**b**) latency vs. privacy-sensitive data ratio of the comparative models used in the study.

Fig. 5 presents the results of evaluating information leakage as the data sampling ratio increases. The PBAD and PAAD models exhibit increased information leakage as the data-sampling ratio increases, whereas the HEAD model maintains negligible leakage across all settings. Although RHAD achieves accuracy and

latency comparable to those of PAAD, it results in higher information leakage because encryption is applied without considering data sensitivity.

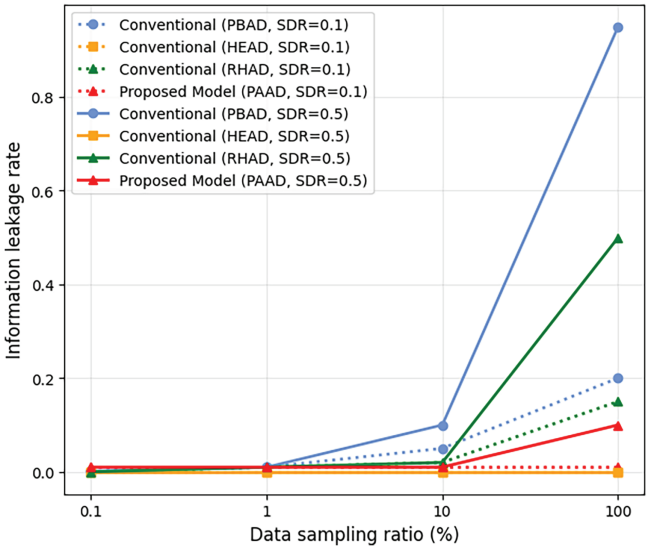


**Figure 5:** Information leakage vs. data sampling ratio of the comparative models used in the study.

Overall, these results demonstrate that PAAD substantially reduces information leakage compared to PBAD, while simultaneously achieving lower latency and higher accuracy than the HEAD model. Furthermore, PAAD effectively preserves privacy even at higher data sampling ratios, which correspond to large-scale network environments with increased traffic volumes.

While the WebAuthAnomalyDetection dataset provides a controlled environment for evaluating PAAD, it is inherently synthetic and may not fully capture the complexity and variability of real-world network traffic. To address concerns regarding realism and potential bias introduced by synthetic data, we additionally evaluate the proposed framework on a real-world anomaly detection dataset.

Fig. 6 illustrates the efficiency comparison of different models under varying privacy-sensitive data ratios. Fig. 6a presents the results obtained on the synthetic WebAuthAnomalyDetection dataset, while Fig. 6b reports the corresponding results on a real-world anomaly detection dataset to assess generalization under realistic traffic conditions.

Across both datasets, the HEAD model consistently exhibits the lowest efficiency. Although the HEAD model provides strong privacy preservation through fully homomorphic inference, its substantial computational overhead results in significantly higher latency and reduced accuracy, leading to poor overall efficiency. When the privacy-sensitive data ratio is low, the PBAD model achieves the highest efficiency, as it maintains high accuracy and low latency, while information leakage remains relatively limited. However, as the proportion of privacy-sensitive data increases, the efficiency of the PBAD model steadily declines, owing to the increasing risk of information leakage.

By contrast, the efficiency of PAAD improves as the privacy-sensitive data ratio increases, reaching a maximum at a certain point. This indicates that adaptively applying homomorphic inference to privacy-sensitive data can effectively balance privacy protection and performance. Beyond this point, the efficiency of the PAAD model begins to decline because the increasing number of homomorphic operations significantly increases latency and reduces accuracy.

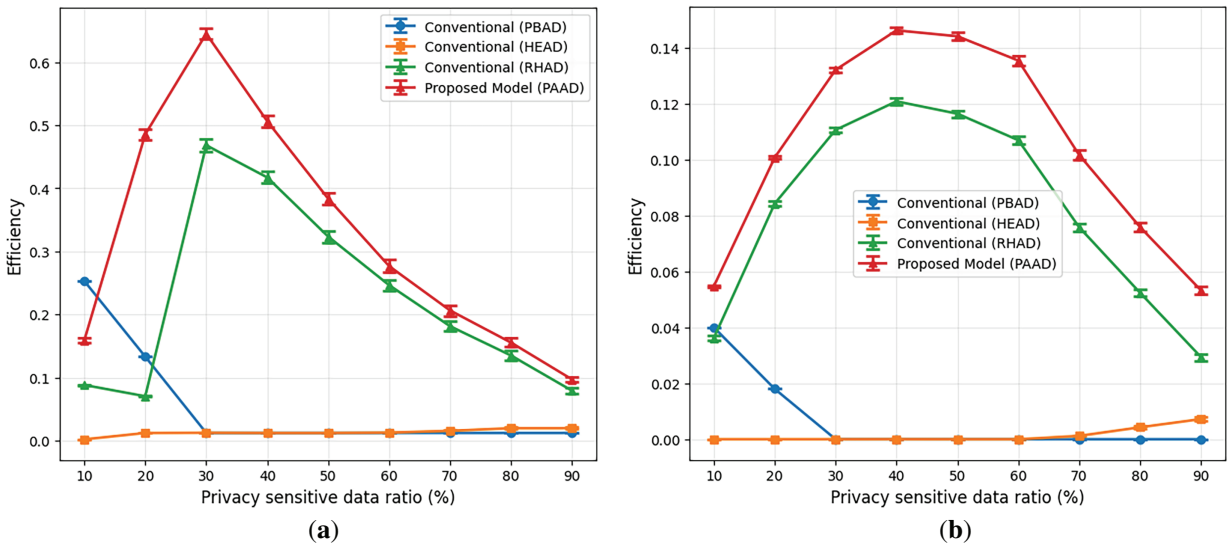


**Figure 6:** Efficiency vs. privacy-sensitive data ratio of the comparative models on (**a**) the synthetic WebAuthAnomalyDetection dataset and (**b**) real-world anomaly detection dataset.

Overall, these results highlight the practical advantages of PAAD compared with conventional anomaly detection systems. First, the findings confirm that while HE is essential for preserving privacy in sensitive traffic, indiscriminate application of encrypted inference can lead to substantial performance degradation in terms of latency and detection accuracy. This trade-off is empirically reflected in the decline in efficiency as the proportion of encrypted traffic increases beyond a certain level. Second, the experimental results demonstrate that PAAD enables researchers to identify an operating point at which privacy protection and system performance are optimally balanced. This indicates that adaptive HE strategies can be effectively tuned to varying traffic conditions, allowing anomaly detection systems to be optimized according to operational priorities without compromising privacy guarantees.

## 6 Conclusion

Privacy-preserving anomaly detection systems are essential for the early identification of security threats across networks and services, while simultaneously safeguarding privacy-sensitive traffic from privacy exposure. Although HE-based anomaly detection techniques can preserve data privacy, they face limitations, including increased computational overhead and reduced detection accuracy. To address these issues, this study proposes an anomaly detection method that dynamically applies HE based on the privacy sensitivity of the data.

To verify the performance of the proposed model, its latency, accuracy, information leakage, and efficiency are evaluated and compared with those of conventional models. The results show that the proposed model reduces latency by a factor of 8.6 compared to a conventional privacy-preserving anomaly detection model, while maintaining an accuracy of 86%, which is slightly lower than that of the plaintext-based model. Furthermore, the proposed model exhibited similar information leakage to the privacy-preserving anomaly detection model and improved the detection efficiency by 111% compared to the plaintext-based model. However, a limitation of the proposed model is that misclassification of privacy sensitivity may lead to information leakage. Future studies will incorporate robustness mechanisms to mitigate potential risks arising from privacy sensitivity misclassification. In addition, we plan to assess the practicality of the

proposed framework on lightweight neural network-based models and realistic deployment hardware, such as edge devices and enterprise servers.

**Acknowledgement:** This paper is a supplementary and extended version of the paper presented at the 9th International Symposium on Mobile Internet Security (MobiSec'25) Conference.

**Funding Statement:** This work was supported by the Ministry of Trade, Industry and Energy (MOTIE) under Training Industrial Security Specialist for High-Tech Industry [grant number RS-2024-00415520] supervised by the Korea Institute for Advancement of Technology (KIAT), Ministry of Science and ICT (MSIT) under the ICAN (ICT Challenge and Advanced Network of HRD) program [grant number IITP-2022-RS-2022-00156310] and National Research Foundation of Korea (NRF) grant [RS-2025-00518150], and the Information Security Core Technology Development program [grant number RS-2024-00437252] supervised by the Institute of Information & Communication Technology Planning & Evaluation (IITP).

**Author Contributions:** The authors confirm their contributions: Yu-Ran Jeon: conceptualization, methodology, software, validation, investigation, resources, data curation, visualization, and writing of the original draft. Seung-Ha Jee: conceptualization, methodology, resources, writing—original draft. Su-Kyoung Kim: methodology, investigation, resources, writing—original draft. Il-Gu Lee: conceptualization, validation, writing—review and editing, supervision, project administration, funding acquisition. All authors reviewed and approved the final version of the manuscript.

**Availability of Data and Materials:** The data supporting the findings of this study are derived from the following public domain resources: WebAuthAnomalyDetection dataset (https://github.com/HamidrezaGholamrezaei/Web_Auth_Anomaly_Detection.git) and real-world anomaly detection dataset (https://www.kaggle.com/datasets/dasgroup/rba-dataset/data).

**Ethics Approval:** Not applicable.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Nomenclature

| | |
|---|---|
| PBAD | Plaintext-based anomaly detection |
| HEAD | Homomorphic encrypted anomaly detection |
| PAAD | Privacy-aware anomaly detection |

## References

1. Jeon SE, Oh YS, Lee YJ, Lee IG. Suboptimal feature selection techniques for effective malicious traffic detection on lightweight devices. Comput Model Eng Sci. 2024;140(2):1669–87. doi:10.32604/cmes.2024.047239.
2. Kil YS, Jeon YR, Lee SJ, Lee IG. Multi-binary classifiers using optimal feature selection for memory-saving intrusion detection systems. Comput Model Eng Sci. 2024;141(2):1473–93. doi:10.32604/cmes.2024.052637.
3. Ruff L, Kauffmann JR, Vandermeulen RA, Montavon G, Samek W, Kloft M, et al. A unifying review of deep and shallow anomaly detection. Proc IEEE. 2021;109(5):756–95. doi:10.1109/JPROC.2021.3052449.
4. Motie S, Raahemi B. Financial fraud detection using graph neural networks: a systematic review. Expert Syst Appl. 2024;240(3):122156. doi:10.1016/j.eswa.2023.122156.
5. Koo K, Park M, Yoon B. A suspicious financial transaction detection model using autoencoder and risk-based approach. IEEE Access. 2024;12(6):68926–39. doi:10.1109/ACCESS.2024.3399824.
6. Li J, Tong X, Liu J, Cheng L. An efficient federated learning system for network intrusion detection. IEEE Syst J. 2023;17(2):2455–64. doi:10.1109/JSYST.2023.3236995.
7. Mehmood M, Amin R, Ali Muslam MM, Xie J, Aldabbas H. Privilege escalation attack detection and mitigation in cloud using machine learning. IEEE Access. 2023;11:46561–76. doi:10.1109/ACCESS.2023.3273895.

8. Wingarz T, See A, Gondesen F, Fischer M. Privacy-preserving network anomaly detection on homomorphically encrypted data. In: Proceedings of the 2024 IEEE Conference on Communications and Network Security (CNS); 2024 Sep 30–Oct 3; Taipei, Taiwan. p. 1–9. doi:10.1109/CNS62487.2024.10735601.
9. Syed NF, Shah SW, Shaghaghi A, Anwar A, Baig Z, Doss R. Zero trust architecture (ZTA): a comprehensive survey. IEEE Access. 2022;10(3):57143–79. doi:10.1109/ACCESS.2022.3174679.
10. Acar A, Aksu H, Uluagac AS, Conti M. A survey on homomorphic encryption schemes: theory and implementation. ACM Comput Surv. 2019;51(4):1–35. doi:10.1145/3214303.
11. Arazzi M, Nicolazzo S, Nocera A. A fully privacy-preserving solution for anomaly detection in IoT using federated learning and homomorphic encryption. Inf Syst Front. 2025;27(1):367–90. doi:10.1007/s10796-023-10443-0.
12. Jeong D, Kim Y, Shin D. Privacy-preserving anomaly detection with homomorphic encryption for industrial control systems in critical infrastructure. IEEE Embed Syst Lett. 2025;17(4):276–9. doi:10.1109/LES.2025.3538013.
13. Shim HY, Park TR, Lee IG. Hybrid encryption technique for low-latency multi-hop communications. In: Proceedings of the 2023 33rd International Telecommunication Networks and Applications Conference; 2023 Nov 29–Dec 1; Melbourne, Australia. p. 252–8. doi:10.1109/ITNAC59571.2023.10368567.
14. Ouyang T, Zhang X. Fuzzy rule-based anomaly detectors construction via information granulation. Inf Sci. 2023;622(1):985–98. doi:10.1016/j.ins.2022.12.011.
15. Rim DN, Heo D, Lee C, Nam S, Yoo JH, Hong JW, et al. Anomaly detection based on system text logs of virtual network functions. Big Data Res. 2024;38:100485. doi:10.1016/j.bdr.2024.100485.
16. Xin R, Liu H, Chen P, Zhao Z. Robust and accurate performance anomaly detection and prediction for cloud applications: a novel ensemble learning-based framework. J Cloud Comput. 2023;12(1):7. doi:10.1186/s13677-022-00383-6.
17. Hooshmand MK, Hosahalli D. Network anomaly detection using deep learning techniques. CAAI Trans Intell Technol. 2022;7(2):228–43. doi:10.1049/cit2.12078.
18. Dong B, Chen D, Wu Y, Tang S, Zhuang Y. FADngs: federated learning for anomaly detection. IEEE Trans Neural Netw Learn Syst. 2025;36(2):2578–92. doi:10.1109/TNNLS.2024.3350660.
19. Lee J, Lee E, Lee JW, Kim Y, Kim YS, No JS. Precise approximation of convolutional neural networks for homomorphically encrypted data. IEEE Access. 2023;11:62062–76. doi:10.1109/ACCESS.2023.3287564.
20. Duc Manh B, Nguyen CH, Thai Hoang D, Nguyen DN, Zeng M, Pham QV. Privacy-preserving cyberattack detection in blockchain-based IoT systems using AI and homomorphic encryption. IEEE Internet Things J. 2025;12(11):16478–92. doi:10.1109/JIOT.2025.3535792.
21. Castro F, Impedovo D, Pirlo G. An efficient and privacy-preserving federated learning approach based on homomorphic encryption. IEEE Open J Comput Soc. 2025;6:336–47. doi:10.1109/OJCS.2025.3536562.
22. Lazea D, Hangan A, Cioara T. Building equi-width histograms on homomorphically encrypted data. Future Internet. 2025;17(6):256. doi:10.3390/fi17060256.
23. Lee JW, Kang H, Lee Y, Choi W, Eom J, Deryabin M, et al. Privacy-preserving machine learning with fully homomorphic encryption for deep neural network. IEEE Access. 2022;10:30039–54. doi:10.1109/ACCESS.2022.3159694.
24. Liu Z, Guo J, Lam KY, Zhao J. Efficient dropout-resilient aggregation for privacy-preserving machine learning. IEEE Trans Inf Forensics Secur. 2023;18:1839–54. doi:10.1109/TIFS.2022.3163592.
25. Idrissi MJ, Alami H, El Mahdaouy A, El Mekki A, Oualil S, Yartaoui Z, et al. Fed-ANIDS: federated learning for anomaly-based network intrusion detection systems. Expert Syst Appl. 2023;234(1):121000. doi:10.1016/j.eswa.2023.121000.
26. Gao S, Ho-Ching Iu H, Erkan U, Simsek C, Toktas A, Cao Y, et al. A 3D memristive cubic map with dual discrete memristors: design, implementation, and application in image encryption. IEEE Trans Circuits Syst Video Technol. 2025;35(8):7706–18. doi:10.1109/TCSVT.2025.3545868.
27. Gao S, Wu R, Iu HH, Erkan U, Cao Y, Li Q, et al. Chaos-based video encryption techniques: a review. Comput Sci Rev. 2025;58:100816. doi:10.1016/j.cosrev.2025.100816.

28. Carlini N, Chien S, Nasr M, Song S, Terzis A, Tramèr F. Membership inference attacks from first principles. In: Proceedings of the 2022 IEEE Symposium on Security and Privacy (SP); 2022 May 22–26; San Francisco, CA, USA. p. 1897–914. doi:10.1109/SP46214.2022.9833649.
29. Rousseeuw PJ, Christmann A. Robustness against separation and outliers in logistic regression. Comput Stat Data Anal. 2003;43(3):315–32. doi:10.1016/S0167-9473(02)00304-3.
30. Benaissa A, Retiat B, Cebere B, Belfedhal AE. TenSEAL: a library for encrypted tensor operations using homomorphic encryption. arXiv:2104.03152. 2021.
31. OpenMined. TenSEAL. [cited 2025 Jan 1]. Available from: https://github.com/OpenMined/TenSEAL.
32. Cheon JH, Kim A, Kim M, Song Y. Homomorphic encryption for arithmetic of approximate numbers. In: Advances in cryptology—ASIACRYPT 2017. Berlin/Heidelberg, Germany: Springer; 2017. p. 409–37. doi:10.1007/978-3-319-70694-8_15.
33. Sharafaldin I, Habibi Lashkari A, Ghorbani AA. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: Proceedings of the 4th International Conference on Information Systems Security and Privacy; 2018 Jan 22–24; Funchal, Portugal. p. 108–16. doi:10.5220/0006639801080116.