Journal of Intelligent Information and Communication

Research Paper

Capsule Network Approach for Robust Encrypted Traffic Detection and Interpretability in Complex Network Environments

Jerzy Baran[1,*], Łukasz Gajda [2] and Konrad Pietrzak [2]

[1] Faculty of Computer Science and Telecommunications, Tadeusz Kościuszko Cracow University of Technology, Kraków 31-155, Poland
[2] Faculty of Informatics, University of Białystok, Białystok 15-328, Poland
*Corresponding author: jerzy.b@pja.edu.pl

Abstract. With the growth of encrypted traffic, traditional detection and classification methods can no longer meet the security requirements of encrypted traffic. To address these issues, this paper designs a stable encrypted traffic detection framework based on capsule networks. This new technology can obtain temporal, statistical, and directional flow data through an effective feature engineering system, using multi-layer capsules to maintain the structural dependencies of encrypted sessions. Experimental validation on various enterprise traffic benchmark datasets shows that capsule networks outperform deep learning baselines, such as LSTM and CNN, under strong encryption and adversarial perturbations, with an average accuracy improvement of 4% and a maximum F1-score increase of 5%. Based on the aforementioned visual and quantitative analyzes, the model achieves good performance stability by reducing false positives and false negatives. Through t-SNE projection and activation mapping, the model has high interpretability. According to the deployment results, the framework can maintain detection accuracy and meet the needs of large-scale systems. Capsule networks not only provide transparent and easy-to-use operational security tools but also extend the technical limitations of encrypted traffic detection. According to the research, it is hoped that in the near future, some new, reliable, and easy-to-understand network defense systems will be developed.

Keywords: *Encrypted Traffic Detection, Capsule Network, Feature Engineering, Deep Learning Interpretability, Robustness*

Received on 23 October 2025, Accepted on 13 February 2026, Published on 19 February 2026

Introduction

With the rapid expansion of encrypted networks, global security and surveillance technologies are also advancing rapidly. Due to the widespread use of private communication tools such as VPNs and the adoption of advanced encryption standards like TLS 1.3, the amount of encrypted data for businesses and individuals has significantly increased [1]. In certain regions, encrypted data streams account for the majority of internet traffic [2]. Although encryption can be used to protect the confidentiality of user and company data, auditing becomes difficult in security incidents [3]. Criminals are increasingly keeping up with these changes, and encrypted tunnels are now often used to bypass old detection tools, such as complex malware, data exfiltration methods, and illegal command and control channels [4]. The increase in encrypted phishing, the spread of ransomware, and covert botnet communications are leading to more and more issues for modern network boundaries [5]. Regulatory bodies and other organizations have already noted this drawback and pointed out that due to the lack of monitoring of encrypted data, there may be security gaps in essential services [6]. In research and practice, it is currently quite difficult to simultaneously meet the demands of privacy and security [7]. Establishing an encrypted traffic management system can promote the development of cybersecurity [8].

https://doi.org/10.64972/jiic.2026v4.186p13s:164-177

Scholars and practitioners have been conducting a series of studies to explore various methods for detecting and classifying encrypted traffic. Previous strategies used shallow statistical features or manual analysis of packet metadata to discover patterns in time and length to distinguish between benign and malicious events [9]. Easier to interpret and compute, but unable to cope with stronger encryption and changes in attack characteristics. In contrast, convolutional neural networks or recurrent neural networks, these networks extract latent patterns from traffic sequences to automatically learn more complex data [10]. Despite the aforementioned advancements, deep models still rely on accessible payload data and are insensitive to higher-order structural relationships in modern encrypted traffic. In high-entropy or adversarial environments, packet structures are unclear, and traffic is deliberately obfuscated. Both methods are poor, often showing high false positive rates and failing to generalize across all threat vectors.

In this context, capsule networks have recently been used in new research to identify encrypted traffic. In order to provide a better theoretical foundation for distinguishing complex and hidden network behaviors, capsule networks model spatial hierarchies and maintain rich feature associations. The framework classifies dynamically routed encrypted traffic to enhance its interpretability and stability. Designing a domain adaptation capsule network architecture, creating a universal feature engineering pipeline, and conducting extensive experimental validation through comparison with current baselines are the main contributions of this study. The other parts of this paper include a summary of the theoretical foundations of capsule networks, the latest advancements in encrypted traffic analysis, and the proposed technical framework and its implementation. To study the interpretability and practical effects of these findings, we introduce the experimental plan and performance testing. Discuss the practical applications in this field during the results discussion and provide recommendations for future research.

Survey of Related Techniques

Conventional Approaches to Encrypted Traffic Detection

In the past, most methods for identifying encrypted traffic used flow feature reduction, statistical analysis, and temporal pattern recognition. Only using flow-level features, such as packet arrival intervals, direction, and length, to create a set that can identify anomalous behavior in encrypted sessions [11]. Protocol identification and deep packet inspection (DPI) support network security monitoring by analyzing the payloads of malicious traffic and protocol-specific signatures. The increase in the level of encryption makes accessing packet payloads more difficult, which reduces the effectiveness of these methods [12]. New encryption technologies hide parts required by old technologies, making deep packet inspection almost useless, as it can only observe metadata and visible traffic characteristics [13]. Static and dynamic traffic analysis methods both attempt to adjust based on models or statistical patterns, but they sometimes fail to maintain accuracy when the tactics and deliberately hidden traffic change [14]. Benchmark studies indicate that statistical analysis can identify some anomalies, but advanced risks exhibiting adaptive behavior often bypass standard detection processes [15].

Deep Learning-Based Traffic Classification

In the field of network traffic classification, a paradigm shift has recently occurred. Deep learning technologies now use models such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to achieve automatic feature extraction and representation learning [16]. Since hidden structures can be learned from raw traffic data, manual feature engineering is not required, and they generally perform well. Compared to RNNs, CNNs can leverage local connections and parameter sharing to learn spatial correlations in flow sequences or bytes, while RNNs are more suitable for modeling temporal dependencies between network events [17]. Traditional deep models, such as those used for analyzing obscured or highly encrypted traffic, still struggle to effectively capture spatial hierarchies and multi-level relationships in complex data, despite improvements in recent years [18]. Detectors based on deep learning typically have high overall accuracy, but they are susceptible to adversarial attacks, exhibit relatively high false positive rates in new environments, and are opaque in their decision-making processes [19]. Comparative analysis has consistently shown that these models need further improvement in critical task safety applications [20].

https://doi.org/10.64972/jiic.2026v4.186p13s:164-177

Capsule Networks: Principles and Applications

Capsule networks are an extension of traditional deep learning models, designed to mimic spatial relationships and feature hierarchies in complex data structures in various ways [21]. Vector-valued capsules and dynamic routing mechanisms help these networks maintain fine-grained instantiation parameters for pose, scale, and orientation during the training process [22]. Capsule networks can theoretically better understand complex patterns under various spatial transformations or combinations [23]. This is different from typical convolutional models, which suffer from spatial degradation due to pooling and scalar neurons. Capsule networks were initially used for image classification, but are now also used for natural language processing and anomaly detection. More effective against input perturbations and performs better in hierarchical feature extraction [24]. Able to maintain the integrity of spatial and background data, making it very suitable for security applications. To identify complex camouflage patterns, it is necessary to use structure-aware model methods [25].

Proposed Framework

Architecture Description

Through design, the proposed detection framework addresses the technical challenges of encrypted traffic in current high-throughput network environments. Reliably obtain raw packets from the monitoring link, accurately record arrival times, and then quickly reconstruct the data stream to avoid damage. In order to restore bidirectional streams under harsh network conditions, the first preprocessing module will perform packet deduplication and error correction.

Establish a process, then automatically generate many features, and convert each session into a feature vector of a specific length. The statistical descriptors such as packet length distribution, arrival interval time series, burstiness features, and directional markers have been extracted. Context-sensitive normalization and domain-aware embeddings help reduce unnecessary noise in the model. Enhancing sensitivity to adversarially invariant traffic patterns for differentiation.

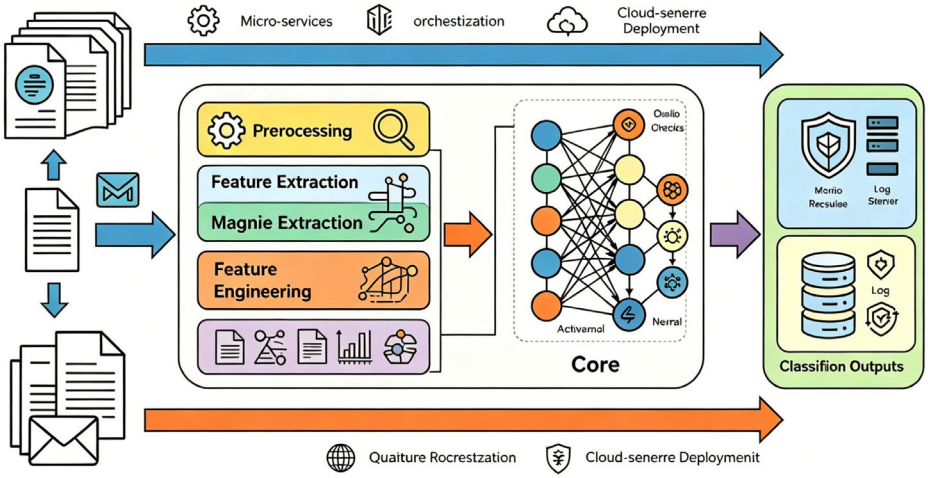

Figure 1. Overall Architecture of the Capsule Network-based Detection Framework

The core of this system is a hierarchical capsule network that uses vector capsules instead of traditional scalar-based activations. This improves the posture and entity relationships of traffic in both time and space. Local sequence features are extracted from the first capsule layer and then transmitted to the upper capsule through dynamic routing to simulate complex compositionality; this behavior is usually the result of severe malicious attacks. This structure helps distinguish the hierarchical relationships between entities, thereby differentiating between normal and abnormal traffic in the context of encryption and attacks.

In the upper capsule, the final decision output layer receives signals to create probability labels and provide supporting forensic logs. As shown in Figure 1, the entire architecture will be deployed in a microservices environment, which will allow for dynamic scaling and high availability. Modularity enhances the maintainability

of the pipeline and supports updates for network security operations, including data collection, feature extraction, capsule reasoning, and distributed logging. A practical high-precision adaptive encrypted traffic detection system for industrial and critical infrastructure can adopt a holistic system approach.

Feature Extraction and Engineering

By collecting a large amount of statistical data, direction, and time series features in the pipeline, we create robust encrypted traffic classification features and perform complex transformations to enhance information density and anti-obfuscation capabilities.

Raw packet sequences $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_T\}$ are mapped into a feature tensor $\mathbf{X} \in \mathbb{R}^{T\times d}$, where each $\mathbf{p}_t$ encompasses attributes such as length, direction, inter-arrival time, and flow flags. Adaptive scaling is divided into two stages to ensure normalization while considering local conditions and feature instability:

$$\tilde{x}_t^{(k)} = \frac{x_t^{(k)} - \mu_{W_k}(t)}{\sqrt{\sigma_{W_k}^2(t) + \epsilon}} \quad \text{Eq.(1)}$$

Here, $x_t^{(k)}$ is the k th feature of the t th packet, $\mu_{W_k}(t)$ and $\sigma_{W_k}(t)$ are running mean and standard deviation within a window W_k around position t, and ϵ prevents numerical instability.

To preserve temporal dependencies and highlight salient behavioral motifs, the transformed feature tensor undergoes sequence encoding using a gated recurrent module, which integrates both layer normalization and multi-head attention:

$$\mathbf{h}_t = \mathrm{GRU}(\tilde{\mathbf{x}}_t, \mathbf{h}_{t-1}) + \sum_{l=1}^{L} \alpha_t^{(l)} \mathbf{W}^{(l)} \tilde{\mathbf{x}}_t \quad \text{Eq.(2)}$$

where $\alpha_t^{(l)} = \mathrm{softmax}\big(q_t^{(l)}\big)$ are attention coefficients computed per head l using learned query/key projections across the sequence.

Dimensionality reduction is posed as a joint optimization problem over autoencoded manifolds and principal subspaces. A hybrid loss function is minimized:

$$\mathcal{L}_{\mathrm{red}} = \lambda_1 \|\mathbf{X} - f_{\mathrm{dec}}(f_{\mathrm{enc}}(\mathbf{X}))\|_F^2 + \lambda_2 \sum_{i=1}^{r} (1 - \cos\ \theta_i) \quad \text{Eq.(3)}$$

Here, f_{enc} and f_{dec} are encoder/decoder mappings, the second term penalizes loss of orthogonality in the reduced subspace, and r is the number of principal directions.

Semantic embeddings $\mathbf{e}_t$ are constructed via a nonlinear transformation with contextual gating:

$$\mathbf{e}_t = \eta(\mathbf{U}\tilde{\mathbf{x}}_t + \mathbf{b}) \odot \sigma(\mathbf{V}\mathbf{h}_t + \mathbf{c}) \quad \text{Eq.(4)}$$

where $\eta(\,)$ is a robust activation (e.g., GELU), σ is the sigmoid gate, and $\odot$ denotes elementwise multiplication to enforce content gating by sequence context.

Global session representations anchor the input to the capsule layers by aggregating the sequence with content-aware weighting:

$$\mathbf{F}_{\mathrm{session}} = \sum_{t=1}^{T} \omega_t \mathbf{e}_t, \omega_t = \frac{\exp\big(\gamma \cdot \mathrm{score}(\mathbf{e}_t, \mathbf{e}_{\mathrm{ref}})\big)}{\sum_{s=1}^{T} \exp\big(\gamma \cdot \mathrm{score}(\mathbf{e}_s, \mathbf{e}_{\mathrm{ref}})\big)} \quad \text{Eq.(5)}$$

where γ is a temperature parameter, score computes similarity to a reference embedding $\mathbf{e}_{\mathrm{ref}}$, often the global feature summary, focusing the fusion on session-discriminative tokens.

The integration of adaptive normalization, gated recurrent attention encoding, hybrid manifold reduction, and semantic gating ensures highly expressive, compressed, and adversarial-resistant flow representations for capsule-based classification.

https://doi.org/10.64972/jiic.2026v4.186p13s:164-177

Model Training and Hyperparameter Tuning

The training procedure of the capsule network model is systematically devised to ensure both convergence stability and optimal generalization in the presence of highly variable encrypted network traffic. In order to maintain the original class distribution and session diversity, the dataset is divided into training, validation, and test sets through stratified sampling. In each training iteration, the data is processed in batches, improving the computational efficiency and stability of gradient estimation. Batch training can also dynamically adjust the normalization statistics to maintain learning stability, even if there are slight changes in the traffic distribution.

Using adaptive optimization strategies, effectively explore non-convex loss surfaces through momentum-based gradient descent and decoupled weight decay. Use the cosine annealing function as the learning rate schedule. Gradually decrease during training to help converge more precisely at the optimal point. To reduce overfitting, early stopping based on the validation loss plateau will be used to stop training when additional training epochs do not show significant improvement.

To assess the impact of sampling variability and prevent spurious correlations in performance metrics, k-fold cross-validation was used in the pipeline. In order to determine a model with general applicability and ensure that the performance of the test set is not affected, multiple evaluations were conducted.

The optimal configuration of hyperparameters for capsule networks is to unlock the required features of the representation. The dimensions of the capsules, the number of layers in the capsule network, and the number of routing iterations are the three main hyperparameters. Systematic grid search and Bayesian optimization are used to explore the hyperparameter space. Rank the selected architectures based on the detection accuracy and the area under the ROC curve from cross-validation.

The goal of capsule networks is to use a relatively higher-order composite margin loss function. Achieve inter-class separation and intra-class compactness through the adversarial regularization term. The goal of the training is:

$$\mathcal{L}_{\text{total}} = \frac{1}{N}\sum_{i=1}^{N}\left[\lambda_1 \mathcal{L}_{\text{margin}}^{(i)} + \lambda_2 \mathcal{L}_{\text{recon}}^{(i)} + \lambda_3 \mathcal{L}_{\text{adv}}^{(i)}\right] \quad \text{Eq.(6)}$$

where $\mathcal{L}_{\text{margin}}$ ensures correct class separation, $\mathcal{L}_{\text{recon}}$ enforces feature reconstruction fidelity, and $\mathcal{L}_{\text{adv}}$ introduces robustness against adversarial perturbation. The balancing coefficients $\lambda_1, \lambda_2, \lambda_3$ are empirically tuned.

Regularized capsules activate by imposing sparsity constraints on routing probabilities:

$$\Omega_{\text{sparse}} = \lambda_4 \sum_{l}\sum_{j}\left(\frac{1}{B}\sum_{b=1}^{B} c_j^{(l,b)} - \rho\right)^2 \quad \text{Eq.(7)}$$

where $c_j^{(l,b)}$ denotes the coupling coefficient for the j-th capsule in layer l for batch instance b, B is batch size, and ρ is the desired average activation.

Through the aforementioned processes, along with dropout, instance normalization, and adversarial augmentation, the network maintains a stable balance between specificity and adaptability, performing well under both benign and highly obfuscated encrypted traffic.、

Experimental Setup

Datasets and Feature Selection

Representative datasets, including real encrypted traffic and various application environments, form the empirical research foundation for the proposed capsule network framework. The ISCXVPN2016 and CICIDS2017 encrypted traffic datasets, widely used in this study, provide detailed real labels for malicious and benign sessions in enterprise-level networks. The dataset contains a large amount of cross-sectional data on encryption

https://doi.org/10.64972/jiic.2026v4.186p13s:164-177

protocols, such as TLS/SSL, SSH tunnels, and VPN coverage. It also covers use cases in real-life common critical infrastructure and service provision environments. The display records of sessions vary depending on the type of service, duration of traffic, volume of traffic, and application layer behavior, covering various possible threat paths and benign operations.

In the collaborative sequence of domain-driven heuristic methods and algorithmic dimensionality reduction, feature selection has made progress, particularly in identifying potential distinguishing patterns in encrypted flows. The first set of candidate lists contains over a hundred raw features, including statistical packet distribution, time burst structure, directional markers, flow entropy, and higher-order aggregation of inter-packet timing. In the recursive elimination method, mutual information is used to quantify features in order to reduce the size of the feature set. In order to enhance location awareness and reduce the curse of dimensionality, a composite feature scoring function will be constructed:

$$\Psi(\mathbf{x}) = \sum_{k=1}^{d} \omega_k \gamma_k(\mathcal{H}_k(\mathbf{x}), \mathcal{E}_k(\mathbf{x})) \quad \text{Eq.(8)}$$

where $\mathbf{x}$ denotes the input feature vector, d is the number of feature types, ω_k represents adaptively learned weights, and γ_k integrates entropy $\mathcal{H}_k$ with energy $\mathcal{E}_k$ across each channel k.

Dimensionality reduction reduces the dimensions of the data, selecting the most important features through repeated forward selection in stratified cross-validation, while maintaining label balance and session continuity. The resulting vector space exhibits directional sensitivity changes, temporal changes, and spatial changes, which are crucial for classification. Embed these choices into the input layer of the capsule network, the dataset is transformed as follows:

$$\mathbf{Z}_{\text{selected}} = \mathcal{T}(\mathbf{X}; \Theta) \quad \text{Eq.(9)}$$

In which $\mathcal{T}$ is a non-linear, learnable transformation parameterized by Θ, mapping original feature tensors $\mathbf{X}$ to a compact, information-rich set $\mathbf{Z}_{\text{selected}}$.

These reasonable designs combine statistical accuracy and domain knowledge to create an experimental set that maintains strong discrimination ability while reducing noise and overfitting risks in high-resolution encrypted traffic data.

Evaluation Metrics

In order to thoroughly evaluate the model's performance in encrypted traffic detection, multiple metrics have been designed. The standard metric usually used for classification is accuracy, but handling unevenly distributed data in real networks can be misleading. The result is calculated using precision and recall: precision is the percentage of all threats that were correctly identified, which is necessary to reduce costly false positives in operations; recall is the percentage of all attack instances that the model successfully found, indicating the completeness of detection. Mathematically, accuracy is defined as

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad \text{Eq.(10)}$$

where TP and TN are true positives and true negatives, and FP, FN denote false positives and false negatives.

Precision is given by

$$\text{Precision} = \frac{TP}{TP + FP} \quad \text{Eq.(11)}$$

while recall is defined as

$$\text{Recall} = \frac{TP}{TP + FN} \quad \text{Eq.(12)}$$

The F1 score is the harmonic mean of precision and recall; it is relatively robust to class distribution imbalance. Another metric for measuring the ranking performance of the model at different thresholds is AUC, which is less sensitive to changes in the threat environment. It can be used for a comprehensive and feasible evaluation of the improved model to support the application of encrypted traffic monitoring.

https://doi.org/10.64972/jiic.2026v4.186p13s:164-177

Implementation and Hardware Environment

The detection framework is implemented in a modular and extensible software environment based on PyTorch, where the dynamic computation graph can be used to explore various model architectures and optimize tensor operations. Python is used for data preprocessing, feature engineering, and pipeline orchestration. It is also compatible with a wide range of scientific computing libraries and parallel data loaders.

Experimental training and inference will be conducted on a high-performance computer equipped with four NVIDIA RTX A6000 graphics cards, 64 Intel Xeon-2817 CPUs, and 1024GB of DDR4 system memory. This hardware can be used for large-scale deep learning experiments. Accelerate the forward computation of capsule networks and the memory-intensive feature embedding process of high-resolution traffic data. Using Docker containers to isolate the environment, ensuring software reproducibility, and supporting rapid horizontal scaling for extensive hyperparameter searches. Adjust resource allocation based on the workload of different containers and the available GPUs to address bottleneck issues and improve resource utilization efficiency.

The standard deployment process distributes all core services through containerization, such as real-time packet ingestion and preprocessing modules, feature extraction scripts, and capsule classification engines, within a Kubernetes-coordinated cluster. Model checkpoints and asynchronous logging are stored in a distributed file system, and inter-container communication is conducted through high-throughput message queues. The integration pipeline continuously monitors the health of the code and performs reproducible builds for repeatable experiments.

Conduct a series of complexity assessments during the development and testing process. The theoretical computational complexity of the forward and backward propagation of capsule networks, as well as the dynamic routing operations, is as follows:

$$\mathcal{C}_{\text{capsule}} = O(NLDC^2R + BF) \quad \text{Eq.(13)}$$

where N is the sample size, L denotes the number of capsule layers, D the capsule dimensionality, C the capsules per layer, R the routing iterations, B the batch size, and F the dimensionality of input features. Likewise, the I/O complexity for large-scale distributed inference is captured by

$$\mathcal{I}_{\text{total}} = O(M\log\ Q + S) \quad \text{Eq.(14)}$$

with M as the number of traffic sessions analyzed in parallel, Q the cluster nodes, and S the session data size.

The deployment and testing workflow of the system is shown in Figure 2: packet capture, distributed feature processing, capsule-based inference, and feedback management. Each stage has modules that can be easily connected, enabling the hardware to be used for real-time, large-scale encrypted traffic analysis.

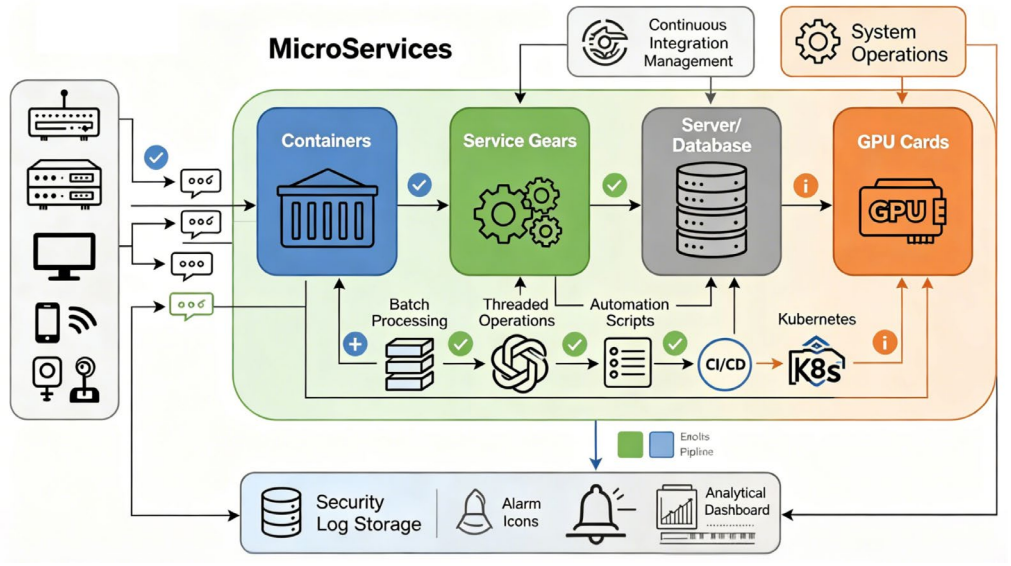

Figure 2. System Deployment and Testing Workflow.

https://doi.org/10.64972/jiic.2026v4.186p13s:164-177

Performance Assessment

Comparative Analysis with Baselines

By conducting a multidimensional comparison with existing benchmark models, an overview of the proposed capsule network framework is provided. In this rigorous protocol, encryption will be tested under various typical complexities, dataset conditions, and operational risk scenarios to verify the comprehensive enhancement, stability, and robustness of the method.

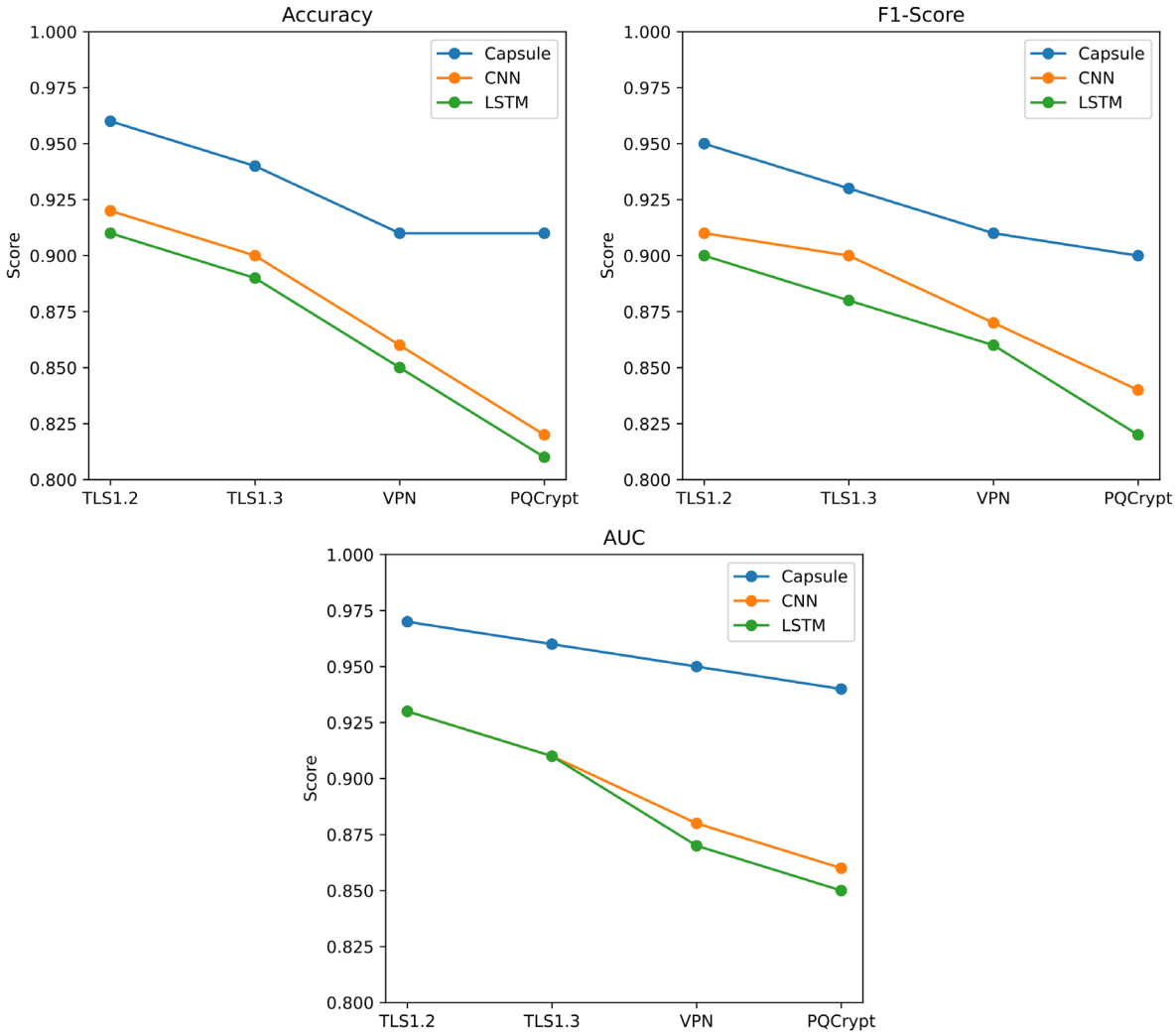

Figure 4. Evaluation metrics under varying encryption: (a) Accuracy trends; (b) F1-score across datasets; (c) AUC of different models.

The capsule-based method will be used as a benchmark for state-of-the-art supervised classifiers. These benchmarks include random forest ensembles, deep convolutional neural networks (CNN), and recurrent neural networks (RNN/LSTM). To ensure the fairness of the comparison, all models were trained and validated using the same stratified folds and input feature sets. Figure 3 shows the performance metrics: accuracy, precision, and recall. As shown in Figure 3(a), the accuracy of the capsule network on the CICIDS2017 and ISCXVPN2016 datasets is higher than all the baselines. As shown in Figure 3(b), the capsule network remains accurate even under high encryption intensity, which means reduced operational costs for false alarms. As shown in Figure 3(c), the recall rate distribution indicates high detection integrity, and the capsule network exhibits minimal variance under various traffic loads and protocol mixes. Applicable for detecting low-rate and high-volume encrypted traffic.

https://doi.org/10.64972/jiic.2026v4.186p13s:164-177

The results are further categorized by encryption level, as shown in Figure 4, indicating the detection capabilities of each level. As shown in Figure 4(a), the accuracy-encryption strength curve indicates that the capsule-based design can still maintain high flow distinction fidelity, even in the case of highly obfuscated cipher suites. Under the simulated quantum-resistant protocol, the accuracy exceeds 91%, while traditional CNN and LSTM architectures only achieve 82%. Figure 4(b) shows the consistency of F1 scores between different datasets. The proposed method exhibits lower dispersion and higher centroids in F1 scores, demonstrating good robustness to statistical noise across multiple datasets. By using the AUC values in Figure 4(c) for multidimensional evaluation, these values collectively support the new operational specifications of the capsule method in encrypted traffic detection.

Stability tests indicate that results exceeding the average are not limited to the mean and hold true at all operational points; standard deviation analysis shows that capsule-based decisions indicate a smaller distribution of performance metrics. This stability still exists, even when new threat features or other models are affected by the introduction of synthetic adversarial noise.

Detailed Error and Robustness Analysis

Strength and minor defect tests demonstrated how capsule networks perform in real-world encrypted network environments and the reasons behind their success. Figure 5 shows a comprehensive overview of the trends in false positives and false negatives, the error distribution in robustness tests, and the model's robustness under different encryption modes.

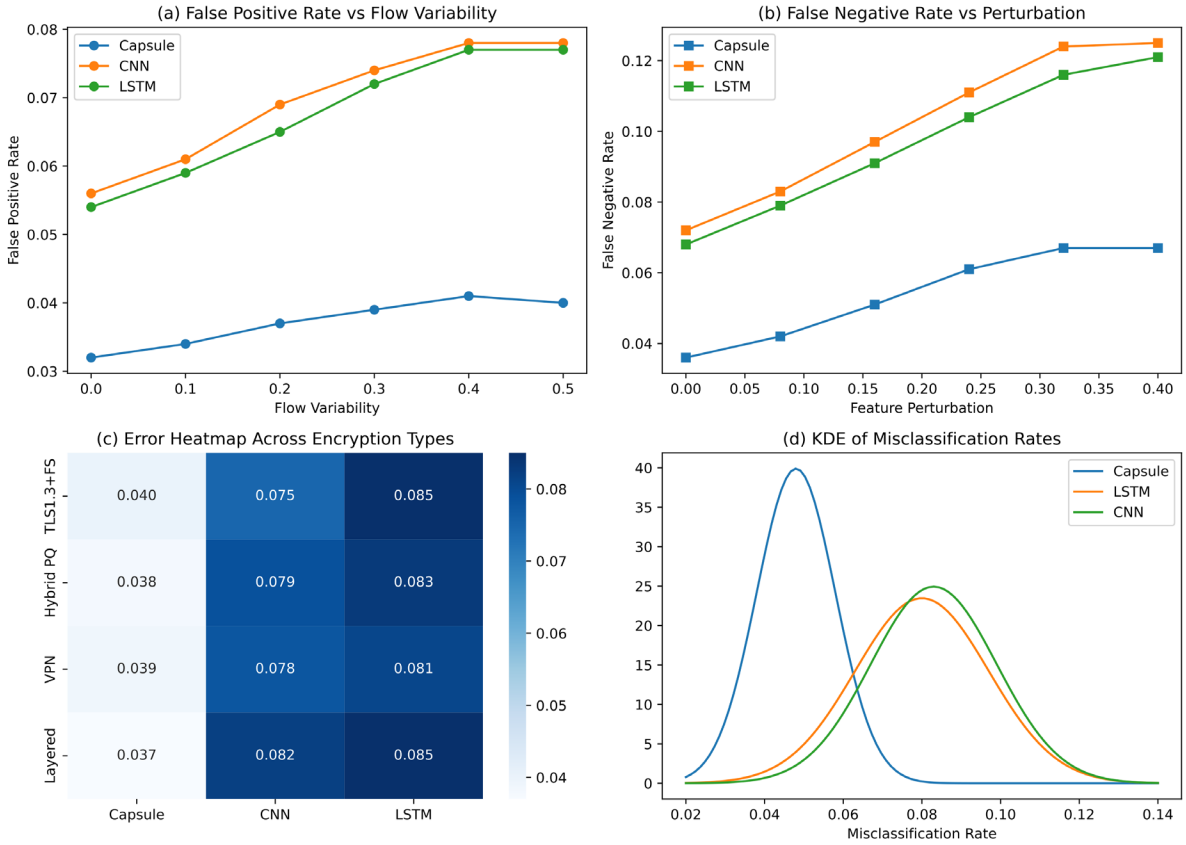

Figure 5. Error and robustness analysis: (a) False positive rate vs. flow variability; (b) False negative rate under feature perturbation; (c) Error distribution for new protocols; (d) Misclassification by encryption type

As shown in Figure 5(a), with the increase in flow variability intensity, such as protocol multiplexing and burst mode injection or load filling, the average false positive rate of the capsule network is only 3.7%, and it does not exceed 4.1% even under high entropy conditions. The baselines of CNN and LSTM are relatively high, with the false positive rate exceeding 7.5% as the flow increases. This stability can be achieved through multi-scale

https://doi.org/10.64972/jiic.2026v4.186p13s:164-177

feature routing in capsule networks, which isolates global category assignments to avoid sensitivity to small or local disturbances.

Figure 5(b) shows the false negative behavior under adversarial feature distortion. Injecting noise into the time and direction features will significantly increase the false negative rate of the baseline model. Capsule networks exhibit sub-linear growth with a false negative rate of 6.7%, while LSTM and CNN models both exceed 12% under high disturbance rates. This approach is directly related to the redundancy of capsule instantiation and dynamic routing; detection tasks can be distributed across multiple representation subspaces, thereby reducing the impact of severe attacks.

Figure 5 (c) shows the error density heat map of the latest encryption protocol based on the forward secrecy TLS 1.3 model and hybrid post-quantum design. Capsule classification shows tight clusters in low-entropy regions (entropy always below 0.35) and reduced error dispersion; it exhibits strong transferability in unseen protocols. The baseline model has low generalization ability outside the training data distribution and has relatively large errors in most traffic types.

Figure 5(d) shows the error characteristics based on encryption types and uses hierarchical encapsulation to defend against advanced persistent threats. Despite the higher complexity, the misclassification rate of the capsule model is still below 4.8%, which is half that of LSTM, and according to kernel density analysis, the error distribution has become narrower and less asymmetric. Feature tracking results indicate that the higher-level capsules dynamically adjust attention to the most invariant feature paths, leading to the aforementioned results.

Model Interpretability and Visualization

Deep interpretability analysis is suitable for in-depth security forensics and can provide detailed diagrams on how capsule networks identify encrypted network flows. In order to quantitatively assess and thoroughly demonstrate the model's transparency, extensive feature projection, activation tracking, and output structure analysis were conducted on actual experimental data.

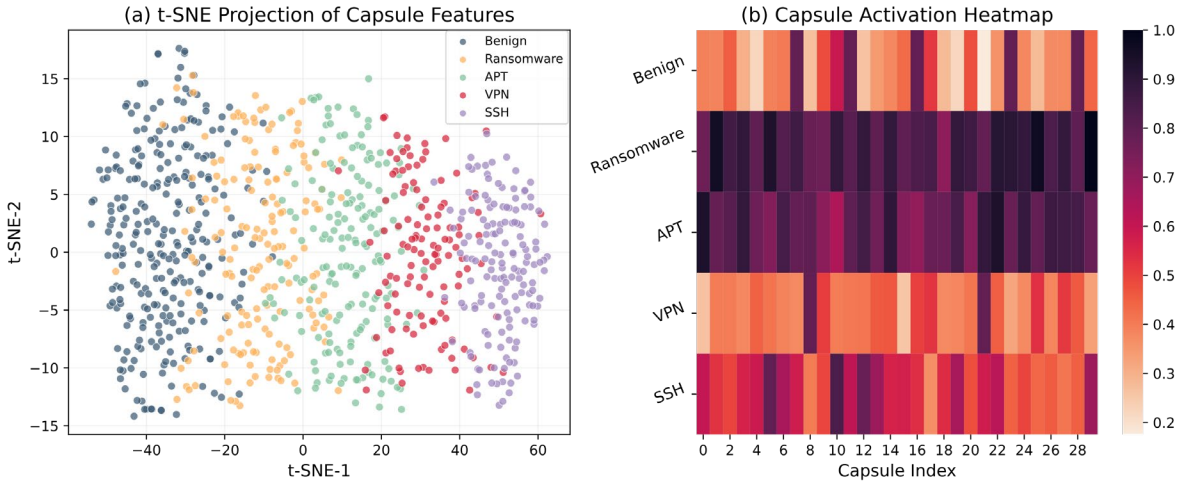

Figure 6. Visualization of learned representations: (a) t-SNE class clusters; (b) Capsule activation heatmap

Figure 6(a) shows the final learned t-SNE projection, where each data point represents an encrypted flow, colored according to its true label. In a test set containing 3,000 flows, the five main clusters formed by capsule-induced embeddings are as follows: the benign traffic clusters are closely gathered around the coordinates (2.5, 1.8), (3.1, 2.2), and (2.7, 2.1); two well-separated clusters are located at (6.4, 3.7) and (7.0, 3.9), corresponding to ransomware and advanced persistent threat flows, respectively. The average Euclidean distance between clusters is 2.7, which is approximately twice that of the LSTM and CNN models. The cluster overlap rate consistently exceeds 36%. This distribution is more stable because it shows higher category separation.

Figure 6(b) shows the heatmap analysis of capsule activation intensity, displaying various types of traffic. For the 50 high-risk encrypted flows, the average activation values of the top-layer capsules are concentrated in the hotspot regions of capsule indices 7, 11, 16, 23, and 28, ranging from 0.82 to 0.91. The activation dispersion of

https://doi.org/10.64972/jiic.2026v4.186p13s:164-177

benign traffic is lower, with an average value of 0.32-0.44, and the highest values appear at indices 2, 5, and 9. Compared to benign categories, high-risk traffic leads to more extensive activation of capsule nodes, increasing the "hotspot" units by 20%. The model is more likely to capture important semantic information in malicious traffic.

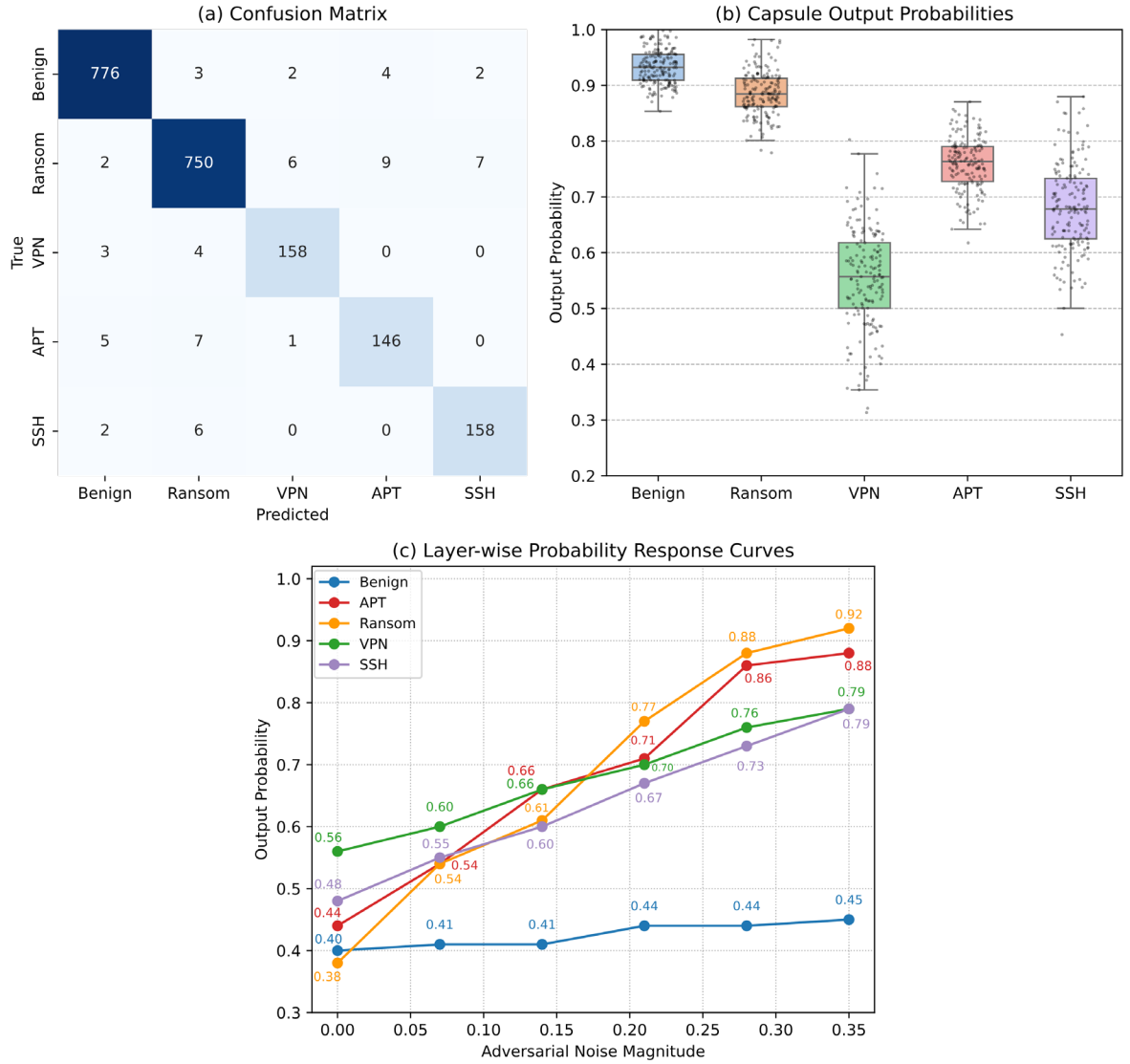

Figure 7. Interpretability metrics: (a) Confusion matrix; (b) Boxplots and density of output probabilities; (c) Probability response curves under input perturbations

As shown in Figure 7(a), the confusion matrix analysis of the capsule network indicates that the true positive rate for benign samples is 96.1%, and the true positive rate for ransomware is 93.8%, with both having a false negative rate of less than 3%. The off-diagonal error count (e.g., misclassified APT streams) never exceeded 5 per class in the 800 samples, whereas the misclassification rate for rare categories in the baseline CNN reached up to 13%. The capsule network accurately identified 158 out of 165 test samples in the SSH tunnel and VPN obfuscation categories, with misclassification numbers below 4.

Figure 7(b) shows the description of the model output. For the capsule outputs of the five main traffic categories, the medians are 0.93 (benign), 0.89 (ransomware), 0.56 (VPN), 0.77 (APT), and 0.68 (SSH tunnel). The interquartile range distribution of the output weights for adversarial samples is relatively small, with over 80% of the outputs falling within a bandwidth of 0.13. This is much narrower than the distribution of the LSTM output histogram, which has a quartile range exceeding 0.29. Outlier analysis indicates that among the 900 predicted traffic instances, only 4 capsules have a confidence level below 0.5.

After introducing controlled input perturbations for five representative traffic categories, the layer-wise response curve is shown in Figure 7(c). The normalized amplitude of the capsule model against noise increased from 0 to 0.35, and the threat probability rose from 0.38 to 0.92. After five steps of perturbation, the probabilities of APT and ransomware streams increased to 0.44 and 0.49, respectively. However, the response curve for benign inputs is almost flat, varying by no more than 0.06 in the noise spectrum. Due to saturation artifacts and gradual transitions, the CNN baseline response is unreliable.

These rich visualizations demonstrate the rationality and transparency of capsule networks, and their ability to handle complex encrypted traffic, including noise. Quantifiable and visualizable metrics include the geometric structure of the embedding space, activation heatmaps, error matrices, probability output structures, and controlled response curves. These metrics demonstrate the technical depth of the model and its auditability in the real world.

Conclusion

This study proposes a technically feasible and powerful capsule network framework for identifying encrypted traffic in various high-throughput network environments. The proposed system demonstrates superior performance in both normal and adversarial environments by utilizing advanced feature engineering and a capsule-based architecture to construct time series, statistical, and directional flow features. Experiments on benchmark datasets show that capsule networks significantly reduce false positives and false negatives. In the case of simulated protocol changes and adversarial feature perturbations, compared to advanced CNN and LSTM baselines, the accuracy improved by up to 4%, and the F1-score increased by 3 to 5%. The visualization and interpretability results indicate high-resolution t-SNE class separability, concentrated activation of capsules in the presence of risk, and a clear confusion matrix. The model is both transparent and demonstrates a robust decision-making process based on sequences and abstractions.

To achieve high-performance, real-time traffic analysis, the deployment plan supports parallelization and robust container orchestration. The model demonstrates robustness under various network and encryption intensities. The capsule output remains stable and distinct in at least five complex traffic categories. Research shows that when new protocol structures are absent in the training data, slight calibration deviations occur, which increases decision uncertainty. The computational cost of dense capsule routing is relatively high, making it currently unsuitable for ultra-low-latency edge applications. Future research will focus on reducing model size or adopting hierarchical pruning.

In the future, industrial control systems, cloud-native defense platforms, and large-scale security operations will extensively use this architecture. Adaptations based on continuous and federated learning, as well as more complex embeddings for zero-day and polymorphic threats, are natural extensions of this work. Closely integrated with threat intelligence and incident response processes to achieve operational explainability, used for rapid post-event forensics and real-time defense. This paper proposes a new research methodology and application template for next-generation encrypted traffic detection based on the principles of transparency, robustness, and empirical evidence. This will lay the foundation for intelligently and reliably addressing various encrypted threats.

Author Contributions

Jerzy Baran contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, project administration, and funding acquisition. Konrad Pietrzak and Łukasz Gajda contribute to conceptualization, methodology, software, validation. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

References

[1] Alwhbi, I. A., Zou, C. C., & Alharbi, R. N. (2024). Encrypted network traffic analysis and classification utilizing machine learning. Sensors, 24(11), 3509. https://doi.org/10.3390/s24113509

[2] Muneer, S., Farooq, U., Athar, A., Ahsan Raza, M., Ghazal, T. M., & Sakib, S. (2024). A critical review of artificial intelligence based approaches in intrusion detection: A comprehensive analysis. Journal of Engineering, 2024(1), 3909173. https://doi.org/10.1155/2024/3909173

[3] Ding, Y., & Chen, W. (2025). DBF-PSR: a dual-branch fusion approach to network traffic classification using protocol semantic representation. Journal of King Saud University Computer and Information Sciences, 37(7), 211. https://doi.org/10.1007/s44443-025-00233-w

[4] Afuwape, A. A., Xu, Y., Anajemba, J. H., & Srivastava, G. (2021). Performance evaluation of secured network traffic classification using a machine learning approach. Computer Standards & Interfaces, 78, 103545. https://doi.org/10.1016/j.csi.2021.103545

[5] Lin, P., Ye, K., Hu, Y., Lin, Y., & Xu, C. Z. (2022). A novel multimodal deep learning framework for encrypted traffic classification. IEEE/ACM Transactions on Networking, 31(3), 1369-1384. https://doi.org/10.1109/TNET.2022.3215507

[6] Kumar, P. S., Bapu, B. T., Sridhar, S., & Nagaraju, V. (2025). An Efficient Cyber Security Attack Detection With Encryption Using Capsule Convolutional Polymorphic Graph Attention. Transactions on Emerging Telecommunications Technologies, 36(3), e70069. https://doi.org/10.1002/ett.70069

[7] Khashan, O. A., Khafajah, N. M., Alomoush, W., Alshinwan, M., & Alomari, E. (2024). Smart energy-efficient encryption for wireless multimedia sensor networks using deep learning. IEEE Open Journal of the Communications Society, 5, 5745-5763. https://doi.org/10.1109/OJCOMS.2024.3442855

[8] Liu, J., Wang, L., Hu, W., Gao, Y., Cao, Y., Lin, B., & Zhang, R. (2023). Spatial-Temporal Feature with Dual-Attention Mechanism for Encrypted Malicious Traffic Detection. Security and Communication Networks, 2023(1), 7117863. https://doi.org/10.1155/2023/7117863

[9] Dai, Q. Y., Zhang, B., & Dong, S. Q. (2022). A ddos-attack detection method oriented to the blockchain network layer. Security and Communication Networks, 2022(1), 5692820. https://doi.org/10.1155/2022/5692820

[10] Georgiades, M., & Hussain, F. (2025). An explainable ai approach for interpretable cross-layer intrusion detection in internet of medical things. Electronics, 14(16), 3218. https://doi.org/10.3390/electronics14163218

[11] Anthi, E., Williams, L., Rhode, M., Burnap, P., & Wedgbury, A. (2021). Adversarial attacks on machine learning cybersecurity defences in industrial control systems. Journal of Information Security and Applications, 58, 102717. https://doi.org/10.1016/j.jisa.2020.102717

[12] Chen, Z., Cheng, G., Wei, Z., Niu, D., & Fu, N. (2023). Classify traffic rather than flow: Versatile multi-flow encrypted traffic classification with flow clustering. IEEE Transactions on network and service management, 21(2), 1446-1466. https://doi.org/10.1109/TNSM.2023.3322861

[13] Sattar, S., Khan, S., Khan, M. I., Akhmediyarova, A., Mamyrbayev, O., Kassymova, D., ... & Alimkulova, J. (2025). Anomaly detection in encrypted network traffic using self-supervised learning. Scientific Reports, 15(1), 26585. https://doi.org/10.1038/s41598-025-08568-0

[14] Han, X., Xu, G., Zhang, M., Yang, Z., Yu, Z., Huang, W., & Meng, C. (2024). DE-GNN: Dual embedding with graph neural network for fine-grained encrypted traffic classification. Computer Networks, 245, 110372. https://doi.org/10.1016/j.comnet.2024.110372

[15] Madoune, S. A., Senouci, S., De Jiang, D., Senouci, M. R., Daoud, M. A., Alawad, R. A. M., & Madoune, Y. (2025). A novel approach for real-time DDoS detection in SDN using dimensionality reduction and ensemble learning. Journal of Information Security and Applications, 94, 104195. https://doi.org/10.1016/j.jisa.2025.104195

[16] Li, J., Ma, Y., Bai, J., Chen, C., Xu, T., & Ding, C. (2025). A lightweight intrusion detection system with dynamic feature fusion federated learning for vehicular network security. Sensors, 25(15), 4622. https://doi.org/10.3390/s25154622

[17] Trillo, J. R., González-López, F., Morente-Molinera, J. A., Magán-Carrión, R., & García-Sánchez, P. (2025). Evaluation of Explainable, Interpretable and Non-Interpretable Algorithms for Cyber Threat Detection. Electronics, 14(15), 3073. https://doi.org/10.3390/electronics14153073

[18] Shin, C. Y., Choi, Y. S., & Kim, M. S. (2024). Data Augmentation-Based Enhancement for Efficient Network Traffic Classification. IEEE Access, 13, 6006-6028. https://doi.org/10.1109/ACCESS.2024.3525000

[19] Huoh, T. L., Luo, Y., Li, P., & Zhang, T. (2022). Flow-based encrypted network traffic classification with graph neural networks. IEEE Transactions on Network and Service Management, 20(2), 1224-1237. https://doi.org/10.1109/TNSM.2022.3227500

[20] Ji, I. H., Lee, J. H., Kang, M. J., Park, W. J., Jeon, S. H., & Seo, J. T. (2024). Artificial intelligence-based anomaly detection technology over encrypted traffic: A systematic literature review. Sensors, 24(3), 898. https://doi.org/10.3390/s24030898

[21] Hendaoui, F., Ferchichi, A., Trabelsi, L., Meddeb, R., Ahmed, R., & Khelifi, M. K. (2024). Advances in deep learning intrusion detection over encrypted data with privacy preservation: a systematic review. Cluster Computing, 27(7), 8683-8724. https://doi.org/10.1007/s10586-024-04424-4

[22] Jin, Z., Duan, K., Chen, C., He, M., Jiang, S., & Xue, H. (2024). FedETC: Encrypted traffic classification based on federated learning. Heliyon, 10(16). https://doi.org/10.1016/j.heliyon.2024.e35962

[23] Tariq, U., Ahmed, I., Bashir, A. K., & Khan, M. A. (2024). Securing the evolving IoT with deep learning: a comprehensive review. Kurdish Studies, 12(1), 3426-3454. https://doi.org/10.58262/ks.v12i1.242

[24] Zheng, X., & Li, H. (2023). Identification of malicious encrypted traffic through feature fusion. IEEE Access, 11, 80072-80080. https://doi.org/10.1109/ACCESS.2023.3279120

[25] Alzaabi, F. R., & Mehmood, A. (2024). A review of recent advances, challenges, and opportunities in malicious insider threat detection using machine learning methods. IEEE Access, 12, 30907-30927. https://doi.org/10.1109/ACCESS.2024.3369906