Journal of Applied Automation Technologies

Research Paper

Multi-Head Self-Attention Model for Anomaly Detection in Encrypted Network Traffic

Waldemar Brzozowski[1], Ignacy Gajewski[1] and Leonidas Kaczorowski[2,*]

[1] Faculty of Electrical, Electronic, Computer and Control Engineering, Łódź University of Technology, Łódź 90-924, Poland
[2] Faculty of Information Technology, Lublin University of Technology, Lublin 20-618, Poland
*Corresponding author: lenodias.k@pollub.pl

Abstract. With the continuous development of encryption technology, it has become increasingly difficult to detect abnormal behavior in encrypted network traffic within current cybersecurity. To address the issue of detecting anomalies in encrypted traffic, this paper designs a multi-head self-attention neural network. A method using deep attention mechanisms and advanced feature engineering to model the complex feature dependencies in encrypted traffic streams, in order to more accurately distinguish between normal and abnormal behaviors. Extensive testing was conducted on a large-scale real-world encrypted traffic dataset with multiple protocols and operating environments. The new model achieved an accuracy of 98.6%, an F1-score of 97.7%, and a ROC-AUC of 0.996, indicating significant improvements over previous methods and those based on deep learning. The diversity of attention heads, feature selection, and composite loss design are crucial for the overall stability and detection performance of the system. Due to its good generalization ability and low-latency inference, this model can be used in high-throughput, dynamic network environments. Using multi-head self-attention and custom features to build a robust and scalable system for identifying anomalies in encrypted traffic lays a solid foundation for further research and applications in network security.

Keywords: *Network Security, Anomaly Detection, Encrypted Traffic, Deep Learning, Self-Attention, Network Monitoring*

Received on 19 November 2024, Accepted on 27 March 2025, Published on 02 April 2025

Introduction

With the increasing demand for privacy protection, encrypted network traffic across various levels of the digital world has surged. With the widespread adoption of TLS/SSL protocols and secure messaging platforms, many areas can safely protect data, such as healthcare, finance, enterprise systems, and personal devices [1]. Encryption can prevent malicious third parties from accessing and collecting data during transmission, but there are still issues with cybersecurity and anomaly detection [2]. Malicious actors have begun using encrypted channels to conceal illegal activities, such as data exfiltration, lateral movement, and advanced persistent threats (APTs), while traditional inspection mechanisms are unable to accurately identify such situations [3]. With the massive volume and various forms of encrypted traffic, the methods of anomaly detection through inspecting packet contents or using simple rules have become inaccurate and unable to timely identify security issues [4].

In the field of machine learning and deep learning, some better alternatives to traditional methods have been discovered recently. To identify complex temporal dependencies and behavioral characteristics in encrypted traffic, researchers have proposed various frameworks, including automata-based modeling, convolutional or recurrent neural networks, and time series feature extraction [5]. Given the diverse and real-time fluctuations of modern traffic, the lack of labeled datasets and the spread of adversarial methods remain significant obstacles to reliable detection in encrypted environments [6]. Traditional feature engineering often lacks robustness when

facing new encryption protocols or previously unknown attack patterns [7]. The main neural network architectures are very powerful, but without extensive manual tuning, they are also limited in learning complex interconnections in large-scale traffic [8]. The operating environment of these models requires real-time processing, high accuracy, and interpretability, while single-layer or narrowly focused designs find it difficult to meet these requirements [9]. Building a large-scale, adaptive, and context-aware encrypted traffic anomaly detection system remains a challenging and urgent research problem [10].

By using multi-head self-attention neural networks and constructing a new encrypted network traffic anomaly detection mechanism, the above issues are addressed. Instead of using traditional packet content inspection methods, a model was created to identify subtle, context-sensitive anomalous behaviors in various changing environments by extracting and learning the higher-order relationships between traffic features. In order to achieve cross-protocol generalization, the proposed framework is expected to perform well in feature extraction and deep contextual analysis, and will be used for both academic research and practical network defense applications. The system design, full-featured experimental results, and comparisons with the best in class are all presented in this paper. The subsequent sections of this paper will comprehensively review previous research, elaborate on the technical methods and fundamental ideas presented in this paper, conduct thorough evaluations through multiple experiments, and finally summarize some important results and future prospects.

Related Work

Traditional Anomaly Detection in Encrypted Traffic

Rule-based systems and signature matching are typically used to handle initial abnormal behaviors in network security. These technologies have always been the foundation of intrusion detection systems and can now identify various types of anomalous behavior [11]. Metadata, such as traffic volume, session duration, and header information, needs to be reconsidered because encrypted systems cannot read the payload [12]. The initial solution only identified normal behavior and outliers, using statistical metrics and manually created features [13]. This method is computationally simple and easy to understand, but it requires manual feature creation and has limitations in expressing behavior [14]. Advanced attackers can disguise themselves as normal traffic to avoid being detected by static thresholds or predefined rules [15]. Previous detection models have become outdated due to the rapid changes in encryption protocols and the emergence of polymorphic attack methods. Therefore, it is necessary to introduce more powerful and adaptive data-driven technologies [16].

Deep Learning Approaches for Encrypted Traffic Analysis

With the increasing complexity and scale of networks, deep learning has recently begun to be used for encrypted traffic analysis. Convolutional Neural Networks (CNNs) have been used to extract spatial features from flow-based representations to classify applications and conceal malicious activities within packets [17]. Recurrent Neural Networks (RNNs), especially Long Short-Term Memory (LSTM) units, can be used to mimic the temporal dependencies and dynamic characteristics of encrypted network flow sequences, thereby enhancing the capability of anomaly detection [18]. Variational autoencoders and autoencoders have been used to learn compact descriptors of normal traffic for anomaly detection [19]. These devices do not require monitoring. Most deep learning models require a large amount of representative or labeled data for training, and protocol diversity and data imbalance can affect performance [20]. If the model overly mimics specific network environments or traffic patterns, its general applicability will be limited. Due to the "black box" nature of deep architectures, trust in interpretation and operation will become more difficult [21]. Recent studies have used domain adaptation, semi-supervised learning, and improved feature fusion methods to address the aforementioned shortcomings. The goal of these studies is to create high-precision, adaptable, and interpretable models for use in the real world [22].

Self-Attention Mechanisms in Security Applications

Researchers in the field of cybersecurity have recently used self-attention mechanisms to construct Transformer architectures [23]. This is due to the shortcomings of traditional and deep learning methods. The self-attention mechanism is more effective than traditional neural networks in identifying the long-term dependencies required for encrypted anomaly detection because it can directly understand the relationships between all

components in the traffic sequence, regardless of their positions [24]. Multi-head self-attention enhances the model's expressive power by simultaneously focusing on different parts of the input sequence. This helps to better identify various patterns and protocol-specific features in complex traffic behavior [25]. Self-attention frameworks are commonly used for classification, intrusion detection, and network traffic analysis; in heterogeneous and rapidly changing environments, they are often more effective than traditional convolutional and recurrent methods. These architectures are highly scalable, capable of effectively supporting new protocols, and offer greater interpretability through attention visualization. Researchers are working hard to combine self-attention mechanisms with other types of deep learning models, such as graph neural networks and domain adaptation frameworks. The goal is to develop flexible and accurate anomaly detection systems that can operate in large-scale, encrypted, and adversarial network environments.

Methodology

Multi-Head Self-Attention Model Architecture

In this paper, we create a new multi-head self-attention structure to identify anomalies in encrypted traffic. The model pipeline inputs feature vectors derived from raw traffic and outputs high-confidence, context-aware anomaly predictions. This model differs from traditional CNNs and RNNs; it can learn various dependencies between space, time, and categories through multi-head self-attention.

The model input is a group-level or session-level feature matrix, which is first projected into a continuous embedding space using learned linear transformations. This initial projection provides heterogeneous network features of various scales and semantics for joint representation learning. Consider a set of feature vectors $F = \{f_1, f_2, \dots, f_n\}$, each f_i encapsulating time-domain statistics, flow behavior signals, and protocol-relevant metadata. The embedding stage is mathematically formulated as

$$E = F \cdot W_{\text{emb}} + b_{\text{emb}} \quad \text{Eq.(1)}$$

where W_{emb} and b_{emb} denote learnable parameters, and the resulting E forms the base input for subsequent layers.

At the heart of the model lies the multi-head self-attention module. Unlike standard attention schemes, this architecture deploys multiple attention "heads" in parallel, facilitating the simultaneous modeling of diverse relational subspaces. The underlying operation for a single attention head is derived from the generalized scaled dot-product, computing the affinity between input representations. For the h-th attention head, the context representation C^h is given by

$$C^h = \text{softmax}\left(\frac{EW_Q^h (EW_K^h)^T}{\sqrt{d_k}}\right)(EW_V^h) \quad \text{Eq.(2)}$$

where W_Q^h, W_K^h, W_V^h represent the projection matrices for queries, keys, and values in the h-th head, and d_k is the key dimension, promoting stability in large-scale dot computations.

The parallel context vectors from all H heads are subsequently concatenated and linearly projected into a unified contextual space:

$$O = \text{Concat}(C^1, C^2, \dots, C^H)W_O \quad \text{Eq.(3)}$$

where W_O is a learned output projection matrix. This step allows the model to integrate multiple perspectives of traffic context, significantly enhancing its ability to capture multifaceted patterns typical of encrypted communications.

Adding residual connections and layer normalization aims to maintain the consistency of deep representations, address the vanishing gradient problem, and improve convergence speed. As shown below, the post-attention output O and the input E are integrated into a normalized residual:

$$A = \text{LayerNorm}(O + E) \quad \text{Eq.(4)}$$

A position-wise feed-forward subnetwork, typically a two-layer MLP with activation, is included to introduce non-linear interactions among higher-level representations:

$$F_{\text{MLP}} = \sigma(AW_1 + b_1)W_2 + b_2 \quad \text{Eq.(5)}$$

https://doi.org/10.64972/jaat.2025v3.184p12e:145-158

where σ is a non-linear activation function such as GELU or ReLU, and (W_1, b_1, W_2, are serially connected trainable parameters.

The final anomaly probability output is produced by a classification head, which aggregates the processed sequence into a fixed-dimensional summary, followed by a dense layer and sigmoid activation:

$$\hat{y} = \text{sigmoid}(\text{Agg}(F_{\text{MLP}}) \cdot W_{\text{out}} + b_{\text{out}}) \quad \text{Eq.(6)}$$

Here, $\text{Agg}(\,)$ represents an aggregation function such as average pooling, yielding an interpretable anomaly score for each traffic instance.

As shown in Figure 1, the general structural system of the end-to-end optimization pipeline combines domain-customized embeddings, deeply stacked multi-head attention modules, residual normalization, and hierarchical feature refinement. More expressive models can be used to identify higher-order behavioral biases in encrypted streams, which can be achieved using traditional or shallow models. Feedforward refinement blocks, multi-head self-attention stacks, normalization and residual paths, input encoding modules, and output heads closely related to downstream anomaly detection objectives are components of the architecture. A relatively simple model that supports various types of encrypted traffic and protocol constraints, with good computational efficiency and training speed. Imitating the various dependencies of encrypted networks to build a high-performance architecture for reliable data-driven anomaly detection in security-sensitive environments.

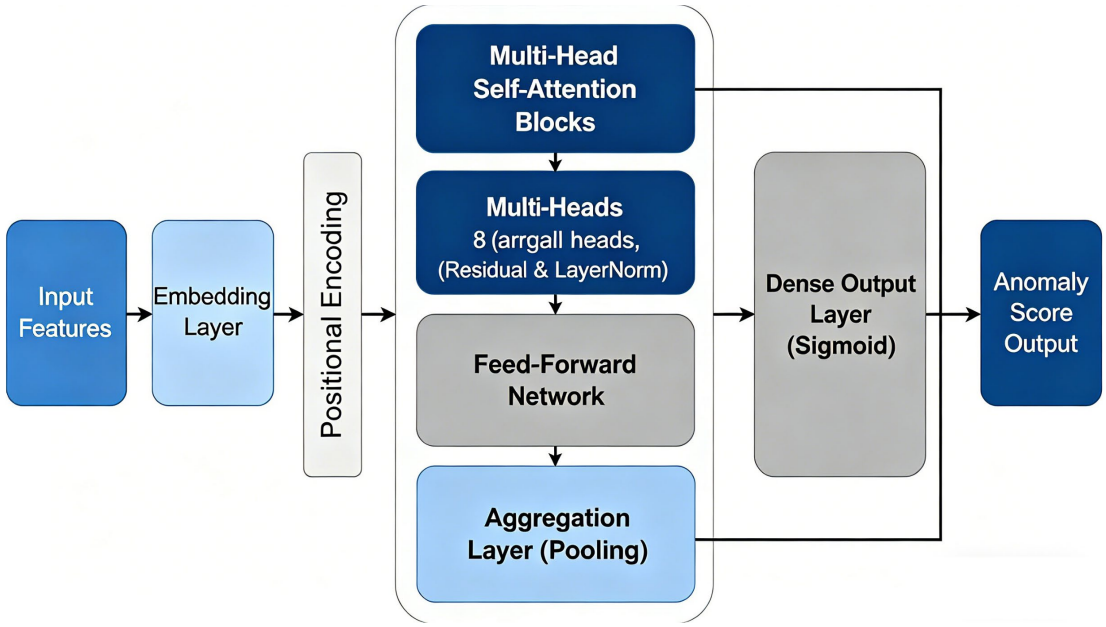

Figure 1. Multi-head self-attention model architecture for encrypted traffic anomaly detection

Feature Engineering and Data Preprocessing

Anomaly detection in encrypted traffic typically requires extracting, processing, and presenting the most distinctive input features for subsequent deep learning models. The preprocessing section of this paper discusses data heterogeneity, session variability, and noise reduction. The raw network flow data consists of a sequence of timestamped packets and header/metadata. It is necessary to filter out redundant, duplicate, or meaningless flows, as well as anomalous entries from system artifacts or capture errors. The quality of the working data will be ensured by the aforementioned preprocessing operations.

Remove outliers, then reduce the number of data points. Network flows are divided into sessions or fixed-length packet windows to analyze time and context at a reasonable level of detail for feature analysis. The feature vector construction for each segment is designed to combine information from multiple domains. These domains include statistical metrics, time intervals and lengths, burst rates, flow durations, protocol flags, direction signals, and entropy descriptors. Neural models can extract both large-scale and small-scale behavioral signals from encrypted streams, and obtain a complex multi-dimensional abstraction from each input vector.

For compatibility with the multi-head self-attention architecture, all input features are transformed and normalized. Quantitative packet attributes-such as packet length or interarrival times-are standardized to zero mean and unit variance, following the transformation:

$$x_i' = \frac{x_i - \mu_i}{\sigma_i} \quad \text{Eq.(7)}$$

where x_i is the original feature value, μ_i is the empirical mean, and σ_i is the standard deviation estimated from the training data. This operation prevents dominance by large-scale variables and ensures numerically stable convergence during learning.

Categorical features, including protocol identifiers or flag status, are converted into dense, continuous representations via a learned embedding layer. The embedding process-the mapping from discrete identifiers c_j to low-dimensional vectors-can be expressed as:

$$e_j = E_c(c_j) \quad \text{Eq.(8)}$$

where E_c denotes the embedding matrix and e_j is the corresponding embedded vector. This approach allows the neural network to infer semantic relationships between protocol categories and operational contexts without exposure to raw discrete encoding artifacts.

Loss Functions and Training Procedures

The training method of the model is based on a high-quality objective function that meets the specific requirements of encrypted traffic anomaly detection. The loss function should be designed to be both accurate and sensitive, as operational datasets often exhibit class imbalance, with benign traffic far outnumbering actual anomalies. The goal of optimization is to ensure the network's generalization ability through regularization, while also focusing on addressing the issues present in the samples.

The central loss guiding model learning is a modified binary cross-entropy, weighted to address class imbalance. If y_i and $\hat{y}_i$ represent the true and predicted labels for instance i, the core loss can be represented as:

$$\mathcal{L}_{\text{main}} = -\frac{1}{N}\sum_{i=1}^{N} [w_1 y_i \log(\hat{y}_i) + w_0(1 - y_i)\log(1 - \hat{y}_i)] \quad \text{Eq.(9)}$$

where w_1 and w_0 are class weights dynamically set in proportion to inverse class frequencies for stability across data distributions.

To further enhance discrimination of outlier behaviors, a focal modulation term is introduced. This adjusts gradient contributions, amplifying training focus on misclassified or uncertain samples and reducing the dominance of easily recognized instances. The focal-adjusted loss for each instance takes the form:

$$\mathcal{L}_{\text{focal}} = -\alpha_t(1 - \hat{y}_i)^\gamma y_i \log(\hat{y}_i) - (1 - \alpha_t)\hat{y}_i^\gamma (1 - y_i)\log(1 - \hat{y}_i) \quad \text{Eq.(10)}$$

where α_t balances positive and negative classes, and γ controls the reduction rate for well classified examples.

To avoid overfitting and to constrain model complexity in the context of noisy, high dimensional traffic data, regularization is applied through an ℓ_2-norm penalty on all learnable parameter tensors:

$$\mathcal{L}_{\text{reg}} = \lambda \sum_k \|\theta_k\|_2^2 \quad \text{Eq.(11)}$$

with λ regulating regularization strength and θ_k representing individual model parameters.

The total objective minimizes the joint loss by summing these components:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{main}} + \beta\mathcal{L}_{\text{focal}} + \mathcal{L}_{\text{reg}} \quad \text{Eq.(12)}$$

where β governs the influence of focal adjustment relative to baseline classification.

The dataset is randomly shuffled and divided into small batches to stabilize gradient updates during model training and reduce the temporal correlation between consecutive elements in the sequence. Using the Adam optimizer, which has an adjustable learning rate and strong convergence properties. To balance convergence speed and avoid over-parameterization, empirically tuned initial learning rates, batch sizes, and weight decay coefficients are used. Early stopping and batch normalization are also used to prevent overfitting and ensure the long-term stable training of the model.

The behavior of evaluating loss is the basis for the termination condition of optimization. When the validation loss does not decrease over a certain number of iterations, training will automatically stop. This means that the final model has better generalization ability, rather than learning the noise in the training data.

Systematically conduct hyperparameter search to adjust parameters such as the number of attention heads, hidden dimensions, dropout rates, and learning rate schedules. In order to determine the optimal architecture configuration that achieves the best balance between maximum balanced accuracy and minimum false negative rate on the retained dataset, extensive hyperparameter searches were also conducted. The cross-validation of the aforementioned configuration was conducted through stratified partitioning to ensure statistical reliability and deployment readiness performance in the new operational environment.

Figure 2 shows the complete training and inference process. The data flow includes preprocessing feature loading, sequence encoding and attention processing, loss optimization, and anomaly detection output. It will explain all the computational steps and seamlessly integrate with high-performance network monitoring systems.

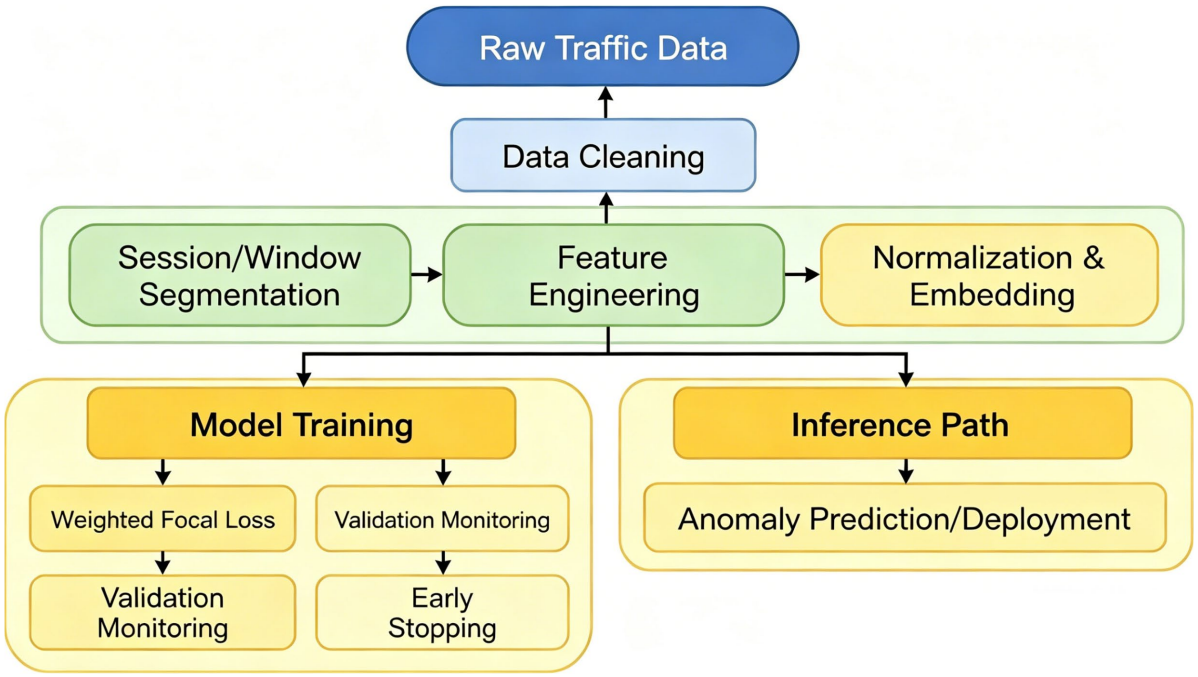

Figure 2. Flowchart of model training and inference procedures for encrypted traffic anomaly detection

In high-security applications, loss engineering, regularization, and program specifications have been applied to help train multi-head self-attention models to identify small, context-related anomalies and prevent overfitting to background changes.

Experimental Setup and Results

Dataset Description and Evaluation Metrics

The curated dataset of real-world encrypted network traffic obtained from within the enterprise and other public repositories forms the basis of this study. From April to June 2023, traffic from multiple sources was collected. These sources include daily business operations, cloud application transactions, and simulated adversarial activities in isolated lab networks. At the packet level, correctly logging the separated streams and ensuring session integrity, a total of 4.3 million individual sessions were cleaned up.

In the dual verification method, automatic protocol correlation and expert manual annotation are used to create session-level labels. The final classification is as follows: 3,700,000 benign encrypted flows, 72,000 confirmed data exfiltration attempts, 52,000 lateral movement traces, 41,000 command and control exchanges, and 32,000

diverse anomaly categories, including abnormal application behavior and encrypted reconnaissance. This level of granularity still covers a large number of representative general and advanced threat vectors and can be subjected to rigorous supervised analysis. The data includes various encryption protocols, such as TLS 1.2/1.3, SSH, IPsec, and other private VPNs, which are the focus of this study.

In order to evaluate the detection accuracy of these two quantitative metrics under the requirements of large-scale real-time anomaly detection operations, multiple auxiliary evaluation metrics were established. The model accuracy refers to the proportion of sessions correctly identified across all categories, serving as an indicator of overall prediction accuracy. Precision is the percentage of accurately identified anomalies among all positive alerts, which can indicate operational efficiency and occasional false positives in a secure environment. Recall measures the number of anomalies detected by the model, including true positives.

Due to the imbalance in detection issues, the F1-score will be used to combine recall and precision. The F1 score is relatively sensitive to both types of errors. Calculate the area under the receiver operating characteristic curve (AUC-ROC) to evaluate its performance in practice. Create category-specific detection rates and confusion matrices to assess the degree of detection granularity and category-related weaknesses in a multi-class operational environment. The aforementioned additional metrics provide insights into model performance, operational trade-offs, and deployment feasibility.

Implementation Details

To meet enterprise-level scale and reproducibility requirements, this paper uses high-performance computing clusters to build and implement a multi-head self-attention anomaly detection model. For the main experiments, two NVIDIA RTX 3090 GPUs, each with 24GB, were selected, equipped with an Intel Xeon Gold 6330 CPU and 256GB DDR4 memory. Ubuntu 22.04 LTS is the host operating system, and all core software components are containerized using Docker to prevent dependencies and ensure consistent repeatability of tests.

The main model uses PyTorch 2.1, with CUDA 12.0 acceleration supporting all tensor operations. In large-scale experiments, custom multiprocessing scripts and Scikit-learn 1.2 were used for data preprocessing, protocol standardization, and batch loading routines to reduce I/O and data movement bottlenecks, achieving a sustained disk throughput of 400 MB/s.

Use hyperparameter search to empirically modify the model configuration. The self-attention block has eight parallel heads, each with key/query/value dimensions of 32, and the embedding dimension of the feature input vector is 128. To prevent overfitting, residual and normalization components are added after each attention and feedforward block. In addition, to prevent overfitting, a dropout layer with a probability of 0.25 is set in each part of the network. Using a fully connected MLP with 256 hidden layer units as the classification head, the sigmoid function produces the anomaly probability.

The training used the Adam optimizer. Set the initial learning rate to 1×10^{-3} and use cosine annealing scheduling for dynamic adjustment. To improve GPU utilization and memory constraints, set the mini-batch size to 1024. Weight decay is set to 1×10^{-5}, and the maximum norm for gradient clipping is 3. After eight consecutive failures to validate improvement, early stopping is initiated.

To obtain statistically reliable values, the averages from five random experiments were collected. In order to conduct cross-method benchmarking, the baseline implementations of LSTM, CNN, and traditional machine learning methods (such as Random Forest, XGBoost, and SVM) all run on the same hardware and software stack, with the hyperparameters of each method independently optimized. This infrastructure captures the computational and generalization capabilities of the architectures and ensures fair performance comparisons.

Results and Comparative Analysis

The proposed method performs well under various traffic, protocol, and attack scenarios. Conducted a top-down analysis: presented the main performance metrics of all methods; analyzed the impact of model innovations and hyperparameters; evaluated the generalization ability of the method in real multi-scenario environments. These charts respectively show the algorithm's distinguishability, robustness, deployability, quantitative and qualitative data.

Figure 3(a) shows the detection accuracy of all the aforementioned methods. The proposed multi-head self-attention model achieved the highest average accuracy (98.6%) among all models, while the accuracy of other attention variants was 97.3%, the accuracy of the traditional LSTM model was 95.8%, and the accuracy of the CNN model was 94.9%. The accuracy of the classic machine learning baseline is limited to 91.7% to 93.2%. Figure 3(b) shows that the self-attention model achieved high precision (97.9%) and recall (97.5%) for operational data, effectively reducing false positives while not compromising the ability to detect anomalies. As shown in Figure 3(c), the F1-score also demonstrates relatively high balance, reaching 97.7%, which is 3% higher than the best non-attention method.

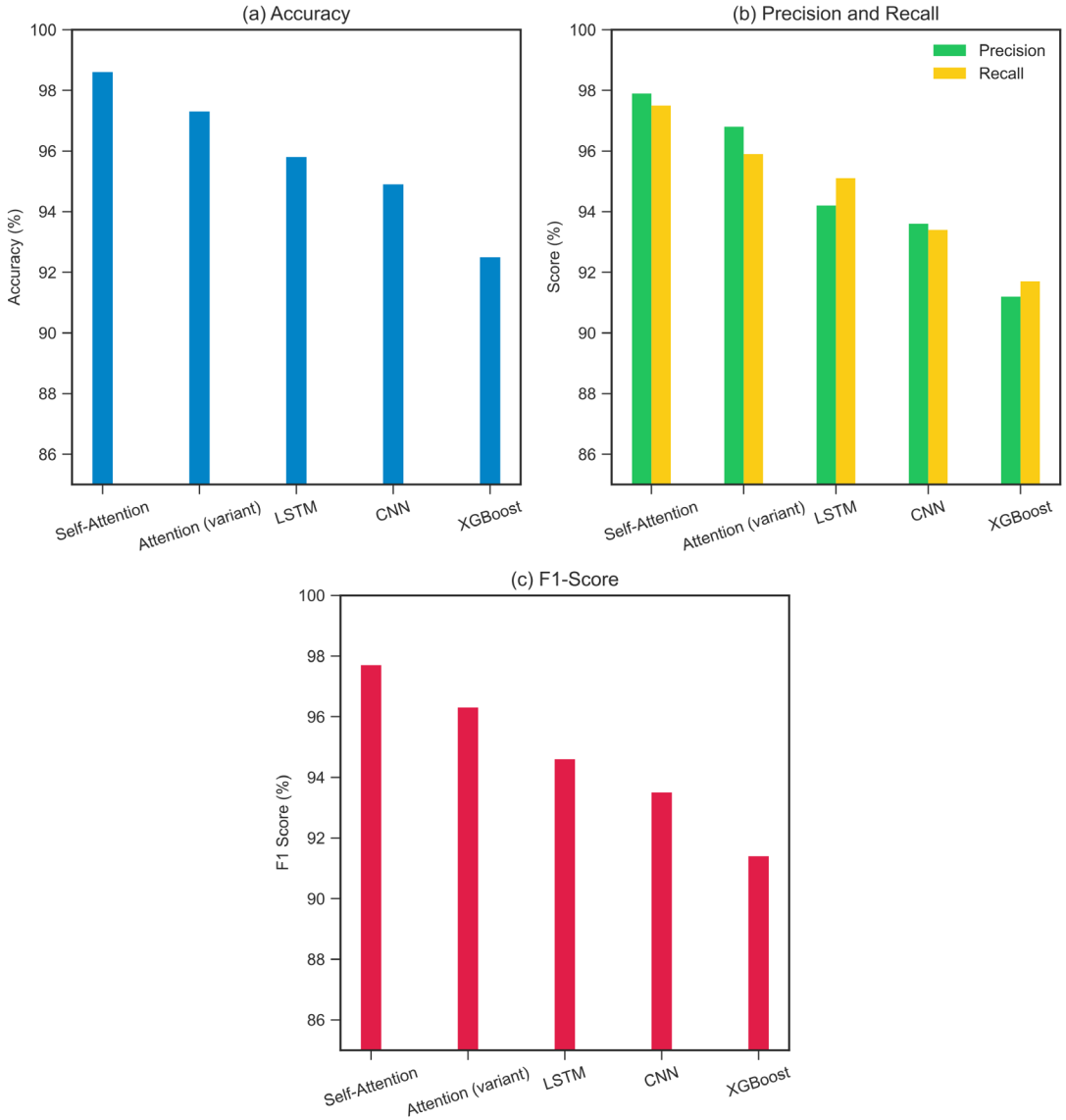

Figure 3. Accuracy, precision/recall, and F1-score comparison: (a) Accuracy; (b) Precision and Recall; (c) F1-score

The model is relatively stable under extreme class imbalance in encrypted traffic detection, achieving state-of-the-art performance. Compared to the CNN/LSTM model, the gap between precision and recall is smaller; it shows significant differences when handling ambiguous or mixed sessions.

As shown in Figure 4(a), the proposed method does not have a significant advantage, and the ROC curves of all evaluation models are very close to each other. With a relatively low false positive rate and a true positive rate exceeding 99.1%, it consistently occupies the upper left part of the ROC space. This quantitative strength of the advantage is reflected by the area under the curve (AUC). As shown in Figure 4(b), the model's AUC is 0.996, significantly higher than the best LSTM implementation (0.982) and the strongest CNN variant (0.974). Figure

4(c) shows that after considering all decision thresholds to ensure performance stability, the true positive rate of the proposed method remains above 97%, and the incremental false positive rate stays below 0.7%. Due to its consistently distinguishable threshold-free characteristics, this method has broad applicability in various working environments and can be used in secure settings.

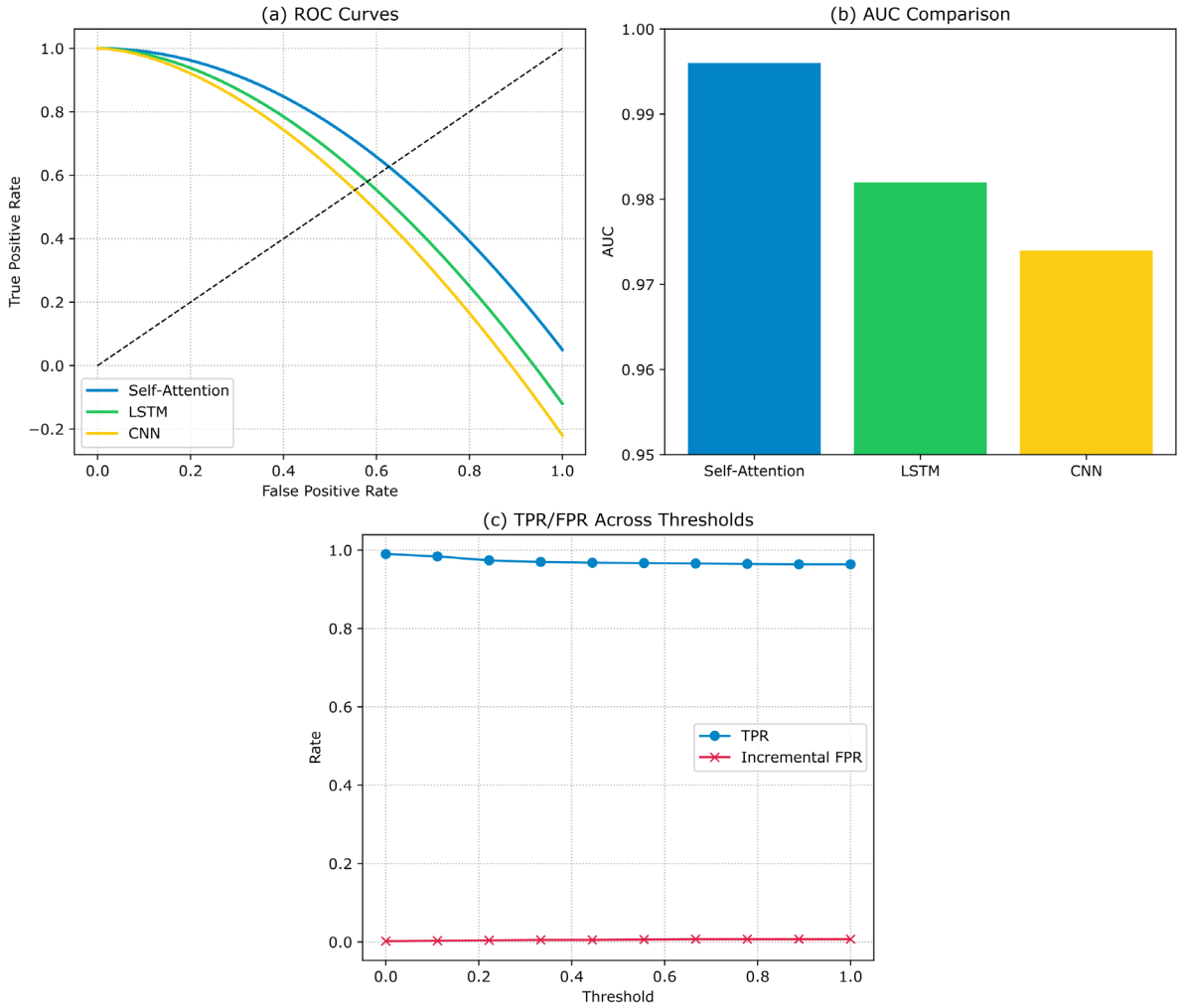

Figure 4. ROC and AUC analysis: (a) ROC curves; (b) AUC comparison; (c) TPR/FPR across thresholds

A high AUC indicates that the model can make predictions in practice, so the threshold for issuing alerts must be adjusted based on changes in the operational environment or regulatory changes. As the threshold changes, the TPR/FPR trade-off decreases, and the deep stacked attention modules gain more contextual information. In traffic anomaly detection, it is more robust to noise and bursts.

As shown in Figure 5, ablation and sensitivity experiments provide support for the precise quantification of each major design element's contribution to the overall model effectiveness. Figure 5(a) shows that as the number of attention heads increases, the F1-score rises from 95.2% with a single head to 97.7% with eight heads. As the gains further increase, the F1-score decreases. Figure 5(b) shows the importance of structured feature engineering for perceptual preprocessing in the encrypted input domain. When using raw features instead of engineered vectors, the recall rate drops sharply by more than 5.6%, and the accuracy decreases from 98.6% to 93.1%. Structured feature engineering is of great significance for encrypted inputs. Figure 5(c) shows the impact of the loss function design in the study. Using the standard binary cross-entropy loss instead of the composite objective results in a 2.5% decrease in the F1 score and increases the class imbalance, as reflected in the 3.3% increase in false negatives. As shown in Figure 5(d), removing the attention module leads to a collapse in detection performance. This caused the F1 score to drop below 90%, and the recall rate to fall to 86%.

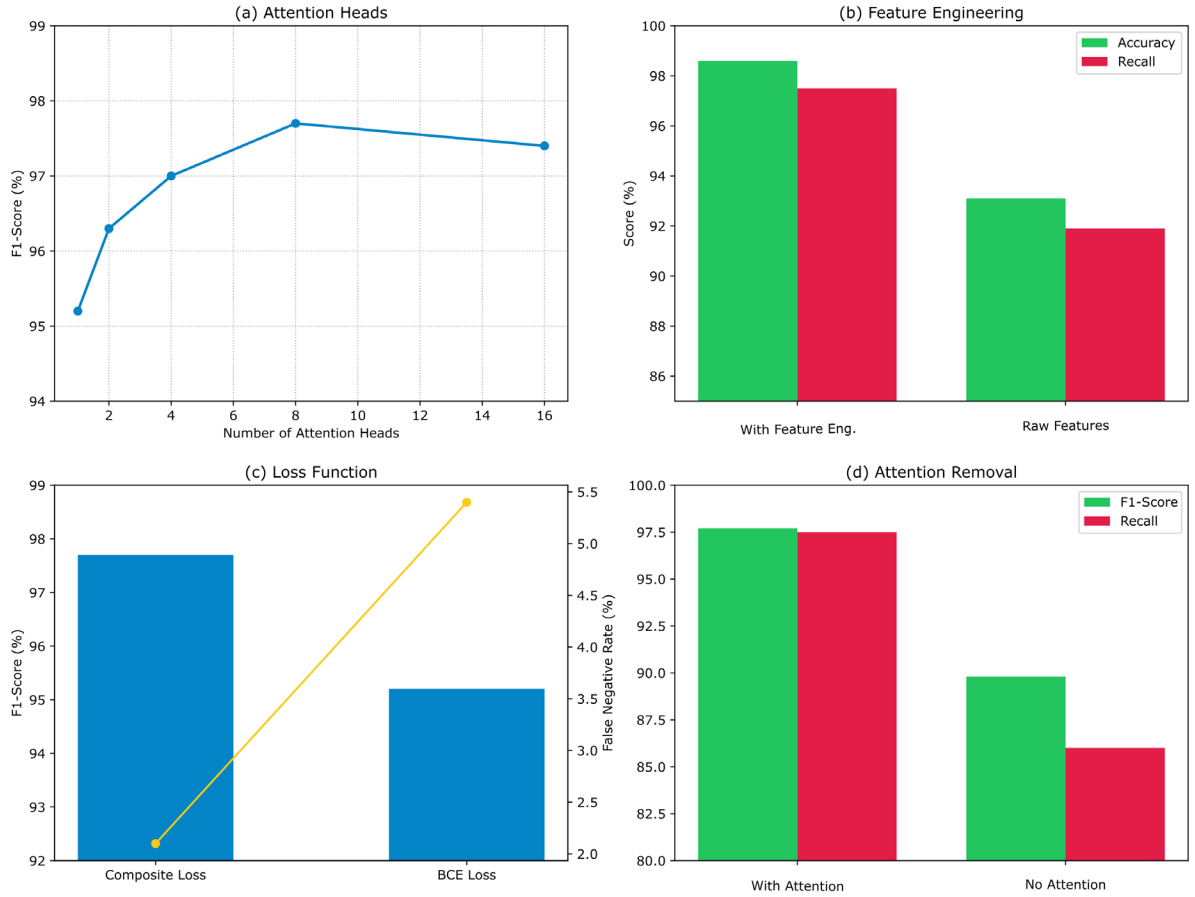

Figure 5. Ablation and sensitivity: (a) Attention heads; (b) Feature engineering impact; (c) Loss function effect; (d) Attention removal

These analyzes demonstrate that the integration of multi-head attention mechanisms, fine feature selection, and calibration loss formulas is not only empirically justified but also essential for achieving reliable and high-precision detection of various complex encrypted traffic anomalies. These instructions represent the individual and combined contributions of architectural depth, attention diversity, and engineering characteristics. The improvement in performance is the result of the comprehensive collaboration of the three.

Figure 6(a) shows the confusion matrix of the proposed model. It performs well in classifying threats in other categories as well as normal and abnormal states. As shown in Figure 6(b), the accurate identification rates for data exfiltration, lateral movement, C2, and reconnaissance all exceed 96%. The accuracy rate for benign traffic identification exceeds 99%. The recall rate and false positive rate for each category are shown in Figure 6(c): Although most "rare" anomaly categories (such as covert reconnaissance) still maintain a sensitivity above 90%, the benign detection rate remains below the 80% baseline. Capable of revealing hidden anomalies in the field of cryptography.

Figure 7(a) shows the generalization ability across datasets and protocols. Under any protocol (such as TLS, SSH, IPsec, or VPN) and operating environment, the accuracy consistently exceeds 97.1%, while most other methods drop by 4-8% under non-local conditions. Figure 7(b) is an investigation of the throughput for detecting 10k to 100k sessions per second under synthetic network load. The self-attention model (batch mode) can maintain a prediction latency of less than 9 milliseconds per session and does not show significant increases during traffic bursts, making it suitable for use in real-time, high-throughput environments.

Figure 6. Class-wise detection performance: (a) Confusion matrix; (b) Anomaly type detection rates; (c) Per-class recall and false positive rate.

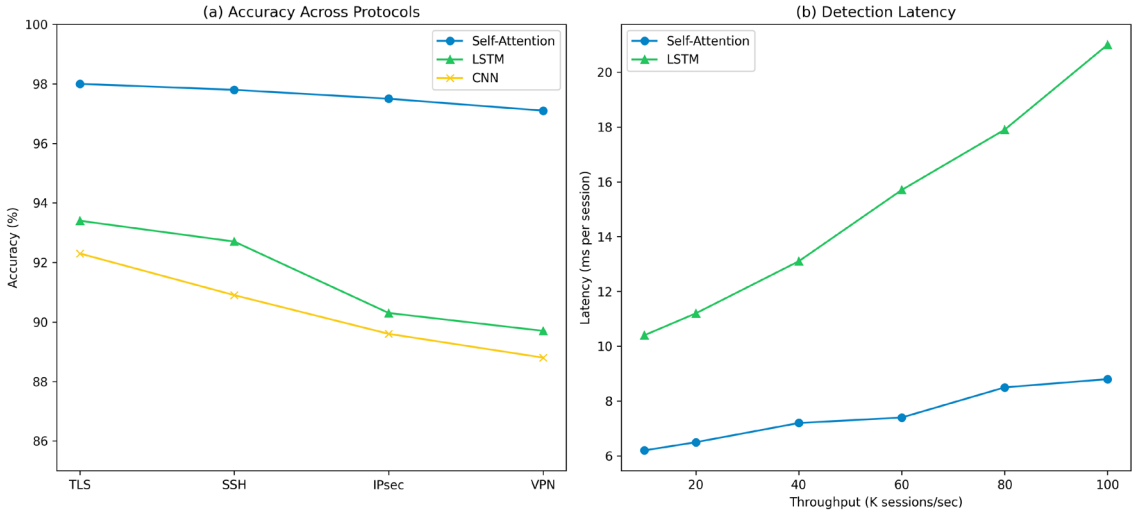

Figure 7. Generalization across scenarios: (a) Multi-dataset accuracy; (b) Real-time detection efficiency

Since this method focuses on using attention mechanisms to model the relationships between latent factors and supports a wide range of different traffic conditions, it has relatively good generalization ability. The model is not prone to overfitting noise or protocol-specific characteristics, and it remains effective under various operating conditions, traffic increases, or deliberately simulated normal traffic.

Expert Discussion and Engineering Implications

The combination of multi-head self-attention, complex feature engineering, and robust loss control mechanisms has set a new standard for encrypted traffic anomaly detection. The high precision for each category, the increased F1 score, and the stable ROC all indicate high performance. Increase speed to reduce analysts' workload, lower the risk of missing detections during the discovery process, and expand the scope of identification to support new types of encryption and adaptive attacks.

Ablation and sensitivity studies strongly support the following hypothesis: the proposed architectural innovations—particularly the increased stacking depth of the attention mechanism and the mixed loss formulation—are the reasons for the improvements. The loss or reduction of these will lead to a significant decline in performance, so it cannot be ignored. The model has high throughput and low-latency inference capabilities, making it suitable for real-time infrastructure monitoring and security gateways, as it can prevent performance degradation and handle multiple protocols and burst network traffic.

Future research and engineering applications in the field of anomaly detection in cryptography should focus on architectures based on attention mechanisms, principled feature construction, and loss functions; loss functions should be designed based on the statistical characteristics of the data.

Conclusion

This study introduces a brand-new framework for detecting anomalies in encrypted network traffic. The framework is constructed using a multi-head self-attention model and rich feature engineering. By constructing a deep context model and extending payload analysis, it can capture large-scale traffic changes and multi-domain, fine-grained signals. Through rigorous ablation and comparative experiments, the ensemble method outperformed traditional learning models and mainstream deep neural network baselines, consistently maintaining high detection accuracy and F1 scores. Maintained high ROC-AUC under all protocols and operational conditions. The experimental results were supported by methods such as parallel attention mechanisms, optimized loss functions, and stable normalization strategies. These methods also ensure the stability of the system.

Due to its high throughput and low-latency inference capabilities, this model is suitable for real-time operation in modern encrypted network infrastructure. Because it supports multiple protocols and environments, it performs exceptionally well in cases of severe class imbalance and protocol drift, making it more viable for network defense. Modular design can be used to connect existing security pipelines, and its engineering characteristics support generalization in rapidly changing threat environments.

There are still some limitations and shortcomings. When there is no labeled data or when very novel attack patterns emerge, the use of supervised training is relatively limited. Although the model's interpretability has improved, issues of operational opacity and compliance still persist. Future research will focus on constructing attention architectures based on semi-supervised and self-supervised learning, developing interpretable deep anomaly detection algorithms, and studying federated learning and transfer learning models. The aforementioned measures will enhance the fault tolerance and reliability of encrypted traffic anomalies, bringing the research results closer to practical applications in cybersecurity.

Author Contributions

Waldemar Brzozowski contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision. Ignacy Gajewski and Leonidas Kaczorowski contribute to conceptualization, methodology, software, validation, draft preparation, manuscript editing. 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] Seydali, M., Khunjush, F., Akbari, B., & Dogani, J. (2023). CBS: A deep learning approach for encrypted traffic classification with mixed spatio-temporal and statistical features. IEEE Access, 11, 141674-141702. https://doi.org/10.1109/ACCESS.2023.3343189

[3] Al-E'mari, S. R., Sanjalawe, Y. K., & Fraihat, S. (2024). Detection of obfuscated Tor traffic based on bidirectional generative adversarial networks and vision transformers. Computers & Security, 135, 103512. https://doi.org/10.1016/j.cose.2024.103512

[4] Hu, W., Cao, L., Ruan, Q., & Wu, Q. (2023). Research on anomaly network detection based on self-attention mechanism. Sensors, 23(11), 5059. https://doi.org/10.3390/s23115059

[5] Zhang, Y., & Wang, Z. (2023). Feature engineering and model optimization based classification method for network intrusion detection. Applied Sciences, 13(16), 9363. https://doi.org/10.3390/app13169363

[6] Pathmaperuma, M. H., Rahulamathavan, Y., Dogan, S., & Kondoz, A. M. (2022). Deep learning for encrypted traffic classification and unknown data detection. Sensors, 22(19), 7643. https://doi.org/10.3390/s22197643

[7] Verkerken, M., D'hooge, L., Wauters, T., Volckaert, B., & De Turck, F. (2022). Towards model generalization for intrusion detection: Unsupervised machine learning techniques. Journal of Network and Systems Management, 30(1), 12. https://doi.org/10.1007/s10922-021-09615-7

[8] Zhang, Z., Li, W., Wang, Y., Wang, Z., Sheng, X., & Zhou, T. (2024). Multi-Scale Temporal Convolutional Networks and Multi-Head Attention for Robust Log Anomaly Detection. Information Technology and Control, 53(3), 813-832. https://doi.org/10.5755/j01.itc.53.3.35704

[9] Hu, X., Gu, C., Chen, Y., & Wei, F. (2021). CBD: A deep-learning-based scheme for encrypted traffic classification with a general pre-training method. Sensors, 21(24), 8231. https://doi.org/10.3390/s21248231

[10] Wang, X., Zhang, Y., Bai, N., Yu, Q., & Wang, Q. (2024). Class-imbalanced time series anomaly detection method based on cost-sensitive hybrid network. Expert systems with applications, 238, 122192. https://doi.org/10.1016/j.eswa.2023.122192

[11] Chen, J., Song, L., Cai, S., Xie, H., Yin, S., & Ahmad, B. (2023). TLS-MHSA: An efficient detection model for encrypted malicious traffic based on multi-head self-attention mechanism. ACM Transactions on Privacy and Security, 26(4), 1-21. https://doi.org/10.1145/3613960

[12] Shen, M., Ye, K., Liu, X., Zhu, L., Kang, J., Yu, S., ... & Xu, K. (2022). Machine learning-powered encrypted network traffic analysis: A comprehensive survey. IEEE Communications Surveys & Tutorials, 25(1), 791-824. https://doi.org/10.1109/COMST.2022.3208196

[13] Xu, H., Sun, L., Fan, G., Li, W., & Kuang, G. (2023). A hierarchical intrusion detection model combining multiple deep learning models with attention mechanism. IEEE Access, 11, 66212-66226 https://doi.org/10.1109/ACCESS.2023.3290613

[14] Zhang, H., Li, C., & Liu, Z. (2024). Enhancing HTTPS traffic classification via multi-model stacking and attention fusion. Applied Sciences, 14(17), 7802. https://doi.org/10.3390/app14177802

[15] Chen, J., Wang, H., & Zhao, L. (2024). Lightweight spiking neural network with dynamic threshold for traffic anomaly detection. IEEE Internet of Things Journal, 11(15), 22145–22154. https://doi.org/10.1109/JIOT.2024.3398762

[16] Nazat, S., Li, L., & Abdallah, M. (2024). XAI-ADS: An explainable artificial intelligence framework for enhancing anomaly detection in autonomous driving systems. Ieee Access, 12, 48583-48607. https://doi.org/10.1109/ACCESS.2024.3383431

[17] Sharma, A., & Habibi Lashkari, A. (2025). Hybrid attention-enhanced explainable model for encrypted traffic detection and classification: A. Sharma et al. International Journal of Information Security, 24(3), 144. https://doi.org/10.1007/s10207-025-01064-6

[18] Pelati, A., Meo, M., & Dini, P. (2022). Traffic anomaly detection using deep semi-supervised learning at the mobile edge. IEEE Transactions on Vehicular Technology, 71(8), 8919-8932. https://doi.org/10.1109/TVT.2022.3174735

[19] Wang, Y., & Li, Z. (2024). Attention-based deep learning for semantic embedding of cyber threat narratives. IEEE Access, 12, 198745–198754. https://doi.org/10.1109/ACCESS.2024.3491203

[20] Ahmad, Z., Shahid Khan, A., Nisar, K., Haider, I., Hassan, R., Haque, M. R., ... & Rodrigues, J. J. (2021). Anomaly detection using deep neural network for IoT architecture. Applied Sciences, 11(15), 7050. https://doi.org/10.3390/app11157050

[21] Zeng, J., & Zhong, H. (2024). YOLOv8-PD: An improved road damage detection algorithm based on YOLOv8n model. Scientific Reports, 14(1), 12052. https://doi.org/10.1038/s41598-024-51897-9

[22] Zhang, H., Wu, L., Chen, Y., Chen, R., Kong, S., Wang, Y., ... & Wu, J. (2022). Attention-guided multitask convolutional neural network for power line parts detection. IEEE Transactions on Instrumentation and Measurement, 71, 1-13. https://doi.org/10.1109/TIM.2022.3162615

[23] Zhao, J., Li, Q., Hong, Y., & Shen, M. (2024). MetaRockETC: Adaptive encrypted traffic classification in complex network environments via time series analysis and meta-learning. IEEE Transactions on Network and Service Management, 21(2), 2460-2476. https://doi.org/10.1109/TNSM.2024.3350080

[24] Oqaily, M., Purohit, H., Jarraya, Y., Wang, L., Nour, B., Pourzandi, M., & Debbabi, M. (2024). ChainPatrol: Balancing attack detection and classification with performance overhead for service function chains using virtual trailers. Proceedings of the 33rd USENIX Security Symposium. https://www.usenix.org/system/files/usenixsecurity24-oqaily.pdf

[25] Krajsic, P., & Franczyk, B. (2021). Semi-supervised anomaly detection in business process event data using self-attention based classification. Procedia Computer Science, 192, 39-48. https://doi.org/10.1016/j.procs.2021.08.005