Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

AE-DSP: An Explainable Intrusion Detection Framework for Zero-Day Attacks Based on SHAP Contribution-Space Modeling

Yifan Liu
Hebei University
Zheng Liu
20257019052@stumail.hbu.edu.cn
Hebei University
Wenlei Chai
Hebei University
Miaolin Wang
Hebei University
Junjie Qiu
Hebei University
Yi Liu
Hebei University

Research Article

Keywords: Zero-day Attack Detection, SHAP Contribution Space, Feature Attribution, Explainable Artificial Intelligence, Network Intrusion Detection

Posted Date: November 10th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11454054/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

AE-DSP: An Explainable Intrusion Detection Framework for Zero-Day Attacks Based on SHAP Contribution-Space Modeling

Abstract

Zero-day attack detection is challenging because target attack classes are unavailable during training, preventing models from directly learning their class-conditional distributions or decision boundaries. Existing methods mainly construct unseen-attack detection criteria from raw traffic attributes, learned representations, classification outputs, or inter-class relationships, while sample-level relationships between input features and model decisions remain underexploited. To address this limitation, we propose AE-DSP, an explainable intrusion detection framework that transforms SHAP contributions from post-hoc explanations into task-relevant decision evidence. Within a unified contribution space, AE-DSP models two complementary forms of anomaly evidence: overall deviation from the normal contribution structure and dimension-wise contribution-deviation patterns represented by Deviation Semantic Prototypes. Because contribution dimensions remain aligned with the original traffic features, detection decisions can also be traced to dominant DeltaSHAP deviations. AE-DSP achieves F1-scores of 0.9341, 0.9250, and 0.9025 on NSL-KDD, UNSW-NB15, and CICIDS2017, with corresponding New Recall values of 0.9774, 0.9350, and 0.9251. Ablation studies, random-seed stability analyses, parameter-sensitivity analyses, and quantitative explainability experiments further demonstrate the effectiveness and complementarity of the proposed contribution-space modeling strategy.

Keywords: Zero-day Attack Detection; SHAP Contribution Space; Feature Attribution; Explainable Artificial Intelligence; Network Intrusion Detection

1 Introduction

Network Intrusion Detection Systems (NIDSs) are essential to modern cybersecurity defense. Learning-based NIDSs have achieved strong performance through ensemble learning (Sayem et al. 2024), optimized machine learning (Injadat et al. 2021), and autoencoder-based anomaly detection (Mirsky et al. 2018). However, their effectiveness depends on whether training data sufficiently cover attack patterns. Closed-set models can learn direct relationships between known attacks and normal traffic, whereas deployed systems may encounter novel attacks or variants with no class information or labeled training samples. The central zero-day challenge is therefore not simply to improve known-attack classification accuracy, but to construct detection criteria from training-time information that generalize to entirely unseen attacks.

Existing zero-day attack detection studies mainly follow anomaly or one-class learning (Hindy et al. 2020; Ruff et al. 2018), open-set recognition (Farrukh et al. 2024; Soltani et al. 2023), and zero-shot or unseen-attack generalization (Rivero et al. 2017; Sarhan et al. 2023).

Although these approaches construct detection criteria from different forms of training-time information, including data distributions, learned representations, classification responses, and class-level relationships, sample-level evidence directly related to the detector decision process remains underexploited.

SHAP decomposes an individual prediction into feature-wise contributions, providing sample-level decision evidence aligned with the original feature dimensions. Existing intrusion-detection studies use feature-contribution information mainly for post-hoc interpretation (Le et al. 2022; Wali et al. 2025), including explanation-guided intrusion-response analysis (Wei et al. 2023), although some work has begun to employ SHAP representations directly for unseen-attack modeling (Barnard et al. 2022). However, a single SHAP vector does not directly indicate how abnormal its decision evidence is relative to normal behavior, and a single overall anomaly score may lose dimension-wise deviation structure. Zero-day modeling therefore requires a normal contribution reference together with overall and dimension-wise deviation modeling.

To address these limitations, we propose AE-DSP, which transforms sample-level SHAP contributions into a unified task-relevant contribution space. Normal traffic is used to establish a contribution reference from which AE-DSP models both overall deviation and dimension-wise contribution-deviation structure. Representative deviation patterns learned from known attacks provide complementary evidence, and the two types of evidence are fused for final detection. Because the contribution dimensions remain aligned with the original traffic features, the resulting decisions can also be traced to dominant contribution deviations.

The main contributions of this work are summarized as follows.

1. We formulate a task-relevant SHAP contribution space for zero-day attack detection, in which the feature-to-decision relationships learned by a supervised detector are explicitly represented at the sample level. The resulting space provides a detector-informed representation for unseen-attack modeling while preserving the correspondence between contribution dimensions and the original traffic features.

2. We develop a normal-reference-driven contribution-deviation modeling strategy that characterizes unseen samples through two complementary forms of anomaly evidence. Overall deviation captures departures from the normal contribution structure, while dimension-wise DeltaSHAP deviations preserve the location, direction, and relative strength of contribution changes and are further related to representative deviation patterns learned only from known attacks.

3. We evaluate AE-DSP on NSL-KDD, UNSW-NB15, and CICIDS2017 under strict held-out protocols through main experiments, class-wise analysis, ablation, random-seed stability, parameter sensitivity, and quantitative explainability. AE-DSP achieves F1-scores of 0.9341, 0.9250, and 0.9025 and New Recall values of 0.9774, 0.9350, and 0.9251, respectively.

Section 2 reviews related work, Section 3 presents AE-DSP, and Section 4 reports the experimental evaluation. Section 5 concludes the paper and discusses limitations and future work.

2 Related Work

2.1 Zero-Day Attack Detection

The main constraint in zero-day attack detection is that the target attack classes are unavailable during training. As a result, the model cannot directly learn their class-conditional distributions or decision boundaries. Existing studies mainly address this problem through anomaly detection and one-class learning, open-set recognition, and zero-shot or unseen-attack generalization. These approaches differ not only in their model architectures and learning objectives. They also differ in the representation spaces used to define unknown attacks and in the evidence used to determine whether a test sample is anomalous or unknown.

Anomaly and one-class methods construct normal-data distributions, learned representations, or compact decision regions, as summarized in deep anomaly-detection literature (Pang et al. 2021). Representative approaches include Deep SVDD (Ruff et al. 2018) and autoencoder-based zero-day intrusion detection (Hindy et al. 2020), which identify attacks according to deviations from normal representations. Because these methods primarily model normal behavior, malicious discriminative information available from known attacks may remain underused.

Open-set recognition provides a general framework for rejecting samples outside known-class support (Bendale and Boult 2016). In network intrusion detection, related approaches employ adaptable open-set modeling, incremental learning, or energy-based rejection criteria (Farrukh et al. 2024; Soltani et al. 2023; Souza et al. 2025). However, unknownness does not necessarily imply maliciousness, and unseen attacks that remain well represented by known classes may produce weak rejection evidence.

Zero-shot and transferable-knowledge approaches exploit reusable attributes or structural relationships between known and unseen attacks. Rivero et al. (2017) represent attack classes on a Grassmann manifold, whereas Sarhan et al. (2023) learn transferable behavioral attributes from known attacks. Their effectiveness depends on reusable attributes, structures, or discriminative relationships between known and unseen attacks.

In summary, different methods rely on different types of detection evidence, and each type has its own information limitations. It is therefore still valuable to further explore the task-relevant discriminative knowledge learned by supervised detectors from normal traffic

and known attacks. Such knowledge can then be transformed into sample-level anomaly representations that are more suitable for detecting unseen attacks.

2.2 Model Contribution for Intrusion Detection and Unknown-Attack Modeling

Existing explainable NIDS studies use feature-contribution analysis for post-hoc detector interpretation (Le et al. 2022; Wali et al. 2025) and explanation-guided intrusion-response support (Wei et al. 2023). However, these methods mainly use contribution information after prediction rather than directly integrating it into unknown-attack anomaly modeling.

Attribution-based unknown-sample modeling moves contribution information beyond post-hoc explanation. Barnard et al. feed SHAP explanations from a supervised detector into an autoencoder for seen and unseen attack detection (Barnard et al. 2022). GAIA models attribution-gradient abnormalities for out-of-distribution detection (Chen et al. 2023), while S&I uses hierarchical attribution and explanation patterns to detect unknown samples (Zhang et al. 2025). These studies establish that learned contribution patterns can be used directly for unknown-sample detection.

Existing attribution-based approaches demonstrate that model contributions can support unknown-sample detection, but they do not fully characterize abnormality relative to normal decision evidence or preserve multidimensional deviation structure. In particular, summarizing contribution changes into an overall anomaly score may lose deviation direction, strength, and cross-dimensional relationships. This motivates modeling both normal-referenced overall deviation and structured dimension-wise contribution deviations.

3 Method

3.1 Framework Overview

AE-DSP organizes sample-level SHAP contributions into a unified task-relevant contribution space and models two complementary types of anomaly evidence: overall deviation from normal decision evidence and the structure of dimension-wise contribution deviations. Normal traffic defines the reference for overall deviation, while representative deviation patterns learned from known attacks characterize structured contribution changes. Held-out attacks remain unseen until testing. Fig. 1 summarizes the overall framework.

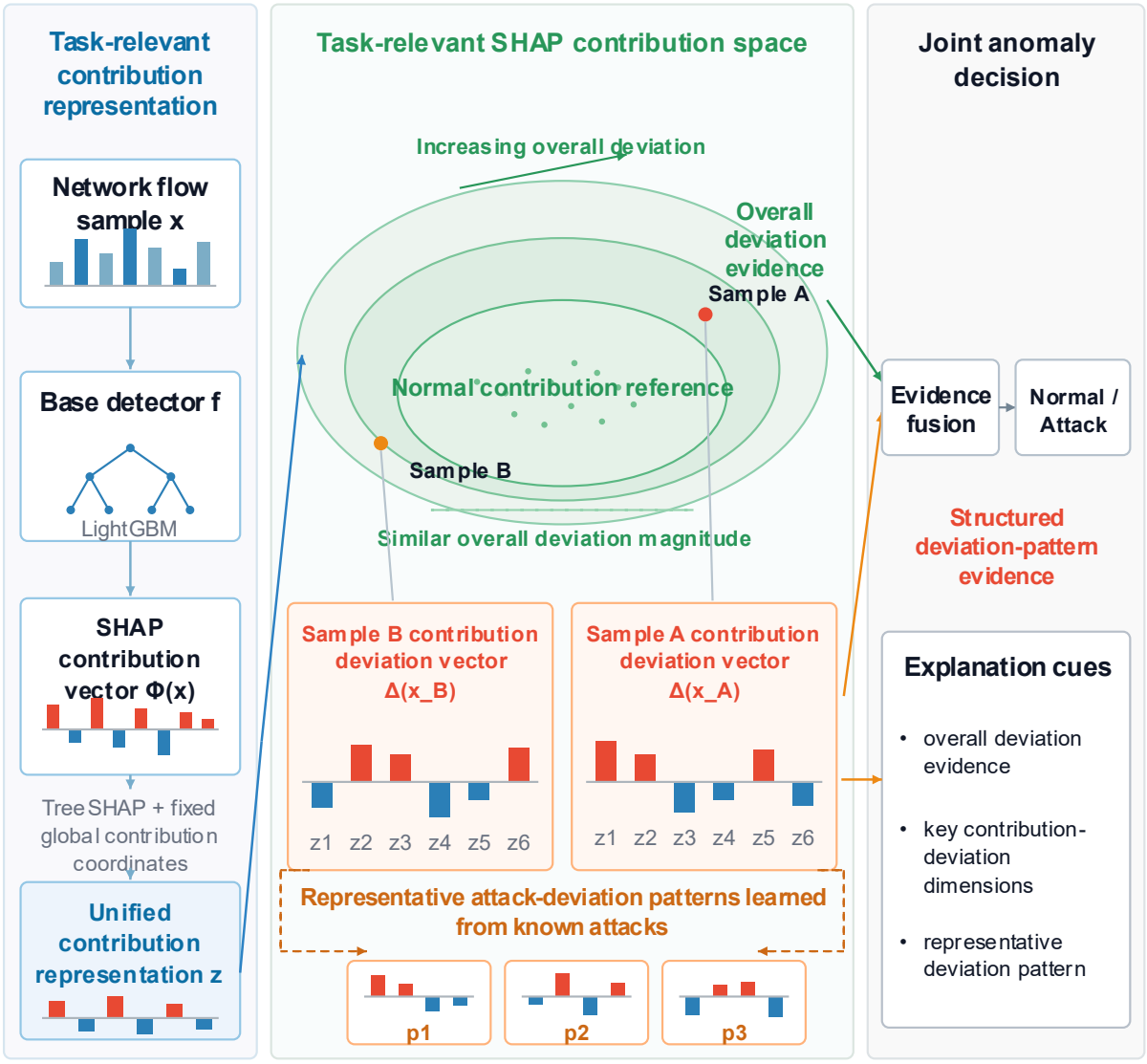


Fig. 1 Contribution-space modeling of AE-DSP for zero-day attack detection

Samples with similar overall deviation levels may still exhibit different DeltaSHAP structures; DSPs preserve this complementary dimension-wise information for the final fusion decision.

3.2 Problem Definition and Held-Out Attack Setting

Let the i-th network traffic sample be denoted by x_i. Its raw input consists of d traffic features. Let the j-th feature be denoted by x_{ij}. The sample x_i can then be represented as shown in Eq. (1).

$$\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{id}]^{\mathrm{T}} \in \mathbb{R}^d \tag{1}$$

Here, d denotes the number of raw traffic features, and $\mathbb{R}^d$ denotes the d-dimensional real-valued feature space.

The final task considered in this study is binary classification between normal and attack traffic. Let y_i denote the binary label of sample x_i. Normal traffic is labeled as 0, while attack traffic is labeled as 1.

$$y_i = \begin{cases} 0, \mathbf{x}_i \text{ is normal traffic}, \\ 1, \mathbf{x}_i \text{ is attack traffic}. \end{cases} \tag{2}$$

Under the held-out attack setting, let $\mathcal{C}_{\text{attack}}$ denote the set of all attack classes, $\mathcal{C}_{\text{known}}$ the set of known attack classes available during training, and $\mathcal{C}_{\text{held}}$ the set of target held-out attack classes that remain unseen during training. The relationship among these sets is defined in Eq. (3).

$$\mathcal{C}_{\text{attack}} = \mathcal{C}_{\text{known}} \cup \mathcal{C}_{\text{held}}, \quad \mathcal{C}_{\text{known}} \cap \mathcal{C}_{\text{held}} = \emptyset \tag{3}$$

Let the training, validation, and test sets be denoted by $\mathcal{D}_{\text{tr}}$, $\mathcal{D}_{\text{val}}$, and $\mathcal{D}_{\text{te}}$, respectively. Let $\mathcal{D}_N$, $\mathcal{D}_K$, and $\mathcal{D}_H$ denote the normal, known-attack, and held-out-attack sample sets, respectively. Both the training set and the validation set contain only normal traffic and known attacks. Their composition is defined in Eq. (4).

$$\mathcal{D}_{\text{tr}} \subseteq \mathcal{D}_N \cup \mathcal{D}_K, \quad \mathcal{D}_{\text{val}} \subseteq \mathcal{D}_N \cup \mathcal{D}_K, \quad \mathcal{D}_H \cap (\mathcal{D}_{\text{tr}} \cup \mathcal{D}_{\text{val}}) = \emptyset \tag{4}$$

During testing, the test set contains normal traffic, known attacks, and held-out attacks. Let $\mathcal{D}_{\text{te}}^N$, $\mathcal{D}_{\text{te}}^K$, and $\mathcal{D}_{\text{te}}^H$ denote the normal, known-attack, and held-out-attack samples in the test set, respectively. The test set is defined as follows.

$$\mathcal{D}_{\text{te}} = \mathcal{D}_{\text{te}}^N \cup \mathcal{D}_{\text{te}}^K \cup \mathcal{D}_{\text{te}}^H \tag{5}$$

Held-out attack samples are excluded from training and validation and are not used for model training, contribution-space construction, anomaly modeling, hyperparameter selection, or threshold calibration. They are introduced only during testing.

3.3 Task-Relevant SHAP Contribution Space

The supervised detector is trained using normal traffic and known attacks. It has therefore learned a task-relevant mapping between raw traffic features and attack-class output. To convert this model-level discriminative knowledge into a sample-level representation, we further decompose the detector output into feature contributions. The contribution of each input feature to the current prediction is then used to construct the model contribution representation of each sample.

Let the base detector be $f(\cdot)$. For the i-th input sample x_i, its malicious prediction output is denoted by $f(x_i)$. LightGBM is used as the base detector in this study. SHAP (Lundberg and Lee 2017) and TreeSHAP (Lundberg et al. 2018) are then used to obtain sample-level contributions corresponding to the raw traffic features. For sample x_i, the SHAP contribution of the j-th raw feature is denoted by ϕ_{ij}. TreeSHAP expresses the detector output as the sum of the baseline output and the contributions of all features.

$$f(\mathbf{x}_i) = \varphi_0 + \sum_{j=1}^{d} \varphi_{ij} \tag{6}$$

Here, ϕ_0 denotes the baseline output of the detector on the background data. The term ϕ_{ij} denotes the SHAP contribution of the j-th raw traffic feature to the attack-class output of sample x_i. The value d is the number of raw features defined in Section 3.2. When $\phi_{ij} > 0$, the feature pushes the model output toward a malicious prediction. When $\phi_{ij} < 0$, the feature contributes in the opposite direction.

After obtaining the feature contributions, the SHAP values of sample x_i across all d raw feature dimensions are combined into a sample-level contribution vector $\boldsymbol{\Phi}(x_i)$.

$$\boldsymbol{\Phi}(\mathbf{x}_i) = [\varphi_{i1}, \varphi_{i2}, \dots, \varphi_{id}]^{\mathrm{T}} \in \mathbb{R}^d \tag{7}$$

Unlike the raw feature vector, the SHAP vector represents feature-wise contributions to the malicious prediction and therefore serves as task-relevant decision evidence.

The complete SHAP contribution vector retains all input dimensions. However, different contribution dimensions have different levels of importance to the detection task. We therefore select a fixed set of contribution dimensions based on the global contribution strength of the samples available during training. This allows the normal contribution structure and deviation patterns learned from known attacks to be modeled in the same fixed coordinates. These coordinates retain the main task-related discriminative information. Let the training set $\mathcal{D}_{\mathrm{tr}}$ contain N_{tr} samples, where $N_{\mathrm{tr}} = |\mathcal{D}_{\mathrm{tr}}|$. For the j-th SHAP contribution dimension, its global importance on the training set is denoted by I_j. It is calculated using the mean absolute SHAP contribution of that dimension.

$$I_j = \frac{1}{N_{\mathrm{tr}}} \sum_{i=1}^{N_{\mathrm{tr}}} \left|\varphi_{ij}\right|, j = 1,2, \dots, d \tag{8}$$

The value I_j measures the overall contribution strength of the j-th dimension among the samples available during training. The term $\left|\phi_{ij}\right|$ denotes the absolute SHAP contribution of the i-th training sample in that dimension. A larger I_j indicates that the corresponding raw feature makes a stronger overall contribution to the detector decisions.

Based on the global importance scores I_j, the k dimensions with the highest scores are selected. They form a fixed Global Top-k contribution index set $\mathcal{S}_k$.

$$\mathcal{S}_k = \mathrm{TopK}\left(\{I_j\}_{j=1}^{d}, k\right) = \{s_1, s_2, \dots, s_k\} \tag{9}$$

The $\mathrm{TopK}(\cdot, k)$ operation selects the indices of the top k dimensions according to global importance. The term s_l denotes the index of the l-th selected contribution dimension. The set $\mathcal{S}_k$ is determined only from samples available during training. It remains fixed during training, validation, and testing. Therefore, held-out attacks are not involved in determining the contribution coordinates.

Based on the fixed index set $\mathcal{S}_k$, the corresponding k dimensions are extracted from the complete contribution vector $\boldsymbol{\Phi}(x_i)$. The resulting task-relevant contribution representation is denoted by z_i.

$$\mathbf{z}_i = \left[\varphi_{i\,s_1}, \varphi_{i\,s_2}, \ldots, \varphi_{i\,s_k}\right]^{\mathrm{T}} \in \mathbb{R}^k \tag{10}$$

The vector z_i is the k-dimensional representation of sample x_i in the unified task-relevant contribution space. Each element ϕ_{is_l} corresponds to the SHAP value of sample x_i at the selected contribution index s_l. Since all samples use the same fixed index set $\mathcal{S}_k$, the same coordinate in z_i always represents the same raw traffic feature and its contribution to the detector decision. Normal samples, known attacks, and held-out attacks can therefore be compared using the same fixed contribution coordinates. The resulting representation z_i is used as the unified input for the following anomaly modeling stages. Section 3.4 models the overall deviation in this space. Section 3.5 further models the structure of dimension-wise contribution deviations.

3.4 Overall Deviation Modeling from Normal Decision Evidence

In the unified contribution space, the contribution behavior of normal training samples provides a task-relevant reference for unseen attack detection. This reference does not depend on any specific unseen attack class. Based on the task-relevant contribution representation z_i constructed in Section 3.3, we use normal training samples to learn the main variation structure of normal decision evidence. The deviation of a test sample from this structure is then used as the overall anomaly evidence.

Let the contribution representations of normal training samples form the normal training set $\mathcal{Z}_N^{\mathrm{tr}}$ in the unified contribution space. According to the definition in Section 3.2, these samples satisfy $y_i = 0$.

$$\mathcal{Z}_N^{\mathrm{tr}} = \{\, \mathbf{z}_i \mid \mathbf{x}_i \in \mathcal{D}_{\mathrm{tr}}, y_i = 0 \,\}, \quad N_N = |\mathcal{Z}_N^{\mathrm{tr}}| \tag{11}$$

This set $\mathcal{Z}_N^{\mathrm{tr}}$ contains the task-relevant contribution representations of all normal training samples. Let N_N denote the number of normal training samples in this set. Since it is constructed only from normal traffic available during training, its contribution behavior serves as the normal decision-evidence reference for overall deviation modeling.

To approximate the main variation structure of normal decision evidence, we use an autoencoder to model $\mathcal{Z}_N^{\mathrm{tr}}$. Let the encoder and decoder be denoted by $\mathrm{Enc}_\theta(\cdot)$ and $\mathrm{Dec}_\psi(\cdot)$, where θ and ψ denote their parameters, respectively. For a normal training contribution representation z_i, the encoder first maps it into a low-dimensional latent representation h_i. The decoder then reconstructs the original contribution representation as $\tilde{z}_i$ from this latent representation.

$$\boldsymbol{h}_i = \mathrm{Enc}_\theta(\boldsymbol{z}_i), \tilde{\boldsymbol{z}}_i = \mathrm{Dec}_\psi(\boldsymbol{h}_i) = \mathrm{Dec}_\psi\big(\mathrm{Enc}_\theta(\boldsymbol{z}_i)\big) \tag{12}$$

The autoencoder is optimized using only normal training samples in $\mathcal{Z}_N^{\mathrm{tr}}$. Its objective is to preserve the contribution structure of normal samples as accurately as possible. The autoencoder training loss is denoted by $\mathcal{L}_{\mathrm{AE}}$, and the mean squared error between the original contribution representations and their reconstructions is used as the training objective.

$$\mathcal{L}_{\mathrm{AE}} = \frac{1}{N_N} \sum_{\boldsymbol{z}_i \in \mathcal{Z}_{\mathrm{tr}}{}^N} \|\boldsymbol{z}_i - \tilde{\boldsymbol{z}}_i\|_2{}^2 \tag{13}$$

After training, for any sample x_i to be detected, its unified contribution representation is z_i and its autoencoder reconstruction is $\tilde{z}_i$. The squared reconstruction error between the original and reconstructed representations is denoted by $e_{\mathrm{rec}}(x_i)$ and is used to measure the raw deviation of the sample from the normal contribution structure.

$$e_{\mathrm{rec}}(\boldsymbol{x}_i) = \|\boldsymbol{z}_i - \tilde{\boldsymbol{z}}_i\|_2{}^2 \tag{14}$$

A larger reconstruction error indicates a stronger deviation from the learned normal contribution structure.

The numerical scale of the raw reconstruction error may vary across datasets and different ranges of normal structure fitting. We therefore apply robust standardization based on the reconstruction error distribution of normal training samples. This provides a common scale for measuring overall deviation relative to normal contribution behavior. The reconstruction errors of all normal training samples are first collected to form the normal reconstruction-error distribution $\mathcal{E}_N$.

$$\mathcal{E}_N = \left\{ e_{\mathrm{rec}}(\boldsymbol{x}_i) | \boldsymbol{z}_i \in \mathcal{Z}_{\mathrm{tr}}{}^N \right\} \tag{15}$$

The distribution $\mathcal{E}_N$ describes how normal training samples deviate from the learned normal contribution structure. It is used as the normal reference for scaling the overall deviation evidence.

For a sample x_i, the robustly standardized overall deviation evidence is denoted by $E_{\mathrm{dev}}(x_i)$ and is defined using the median and interquartile range of the normal reconstruction-error distribution.

$$E_{\text{dev}}(\boldsymbol{x}_i) = \frac{e_{\text{rec}}(\boldsymbol{x}_i) - \text{median}(\mathcal{E}_N)}{\text{IQR}(\mathcal{E}_N) + \varepsilon} \tag{16}$$

The term $\text{median}(\mathcal{E}_N)$ represents the median of the normal training reconstruction errors, $\text{IQR}(\mathcal{E}_N)$ represents their interquartile range, and ε is a small positive constant used to avoid division by zero. A larger $E_{\text{dev}}(x_i)$ means that the reconstruction error is farther from the normal training distribution. It also means that the sample is more difficult to explain using the learned normal contribution structure.

Because the overall score compresses multidimensional contribution changes, Section 3.5 further preserves their dimension-wise structure.

3.5 Dimension-Wise Contribution Deviation Modeling

In the unified contribution space, we use the dimension-wise contribution behavior of normal samples as a reference. This allows the contribution changes of any sample relative to normal behavior to be represented in a standardized form. Based on the normal training representations $\mathcal{Z}_N^{\text{tr}}$ defined in Section 3.4, we construct a normal contribution baseline and a corresponding scale for each contribution dimension. The resulting dimension-wise contribution-deviation representation is referred to as DeltaSHAP.

First, the mean contribution of normal training samples in each dimension is used to construct the normal contribution baseline $\boldsymbol{\mu}_N$.

$$\boldsymbol{\mu}_N = \frac{1}{N_N} \sum_{\boldsymbol{z}_i \in \mathcal{Z}_{\text{tr}}{}^N} \boldsymbol{z}_i \tag{17}$$

Here, N_N denotes the number of normal training samples defined in Section 3.4, and $\boldsymbol{\mu}_N \in \mathbb{R}^k$ represents the average contribution pattern of normal training samples in the unified contribution space. Its j-th component describes the average decision contribution of normal traffic in the j-th contribution dimension. It serves as the reference for the following dimension-wise deviation calculation.

To reduce scale differences across contribution dimensions, we further calculate the standard deviation vector $\boldsymbol{\sigma}_N$ of normal training samples across all contribution dimensions.

$$\boldsymbol{\sigma}_N = \sqrt{\frac{1}{N_N} \sum_{\boldsymbol{z}_i \in \mathcal{Z}_{\text{tr}}{}^N} (\boldsymbol{z}_i - \boldsymbol{\mu}_N)^{\odot 2}} \tag{18}$$

The vector $\boldsymbol{\sigma}_N \in \mathbb{R}^k$ describes the natural variation of normal training samples across different contribution dimensions. The term $(\cdot)^{\odot 2}$ denotes element-wise squaring, and the square-root operation in Eq. (18) is also performed element by element.

For any sample x_i, its unified task-relevant contribution representation is z_i. Its standardized dimension-wise deviation from the normal contribution baseline is denoted by Δz_i, which is referred to as DeltaSHAP.

$$\Delta \mathbf{z}_i = \frac{\mathbf{z}_i - \boldsymbol{\mu}_N}{\boldsymbol{\sigma}_N + \varepsilon} \tag{19}$$

The subtraction and division in Eq. (19) are performed separately for each contribution dimension, so $\Delta z_i \in \mathbb{R}^k$. The term ε is a small positive constant used to avoid division by zero.

DeltaSHAP is not a difference between raw traffic features. Instead, it describes how task-relevant malicious decision evidence deviates from the normal contribution baseline. Let Δz_{ij} denote the j-th component of Δz_i. When $\Delta z_{ij} > 0$, that dimension contributes more strongly toward the malicious prediction than the normal baseline. When $\Delta z_{ij} < 0$, its contribution shifts in the opposite direction. A larger absolute value of $|\Delta z_{ij}|$ indicates a greater deviation from normal contribution behavior. Therefore, DeltaSHAP preserves the location, direction, and relative strength of abnormal contribution changes.

After obtaining the DeltaSHAP representations, we extract the dimension-wise contribution deviations of known attacks available during training. The DeltaSHAP vectors of all known attack samples form the training set of attack contribution deviations $\mathcal{A}^{\Delta}_{\text{known}}$.

$$\mathcal{A}^{\Delta}_{\text{known}} = \{\, \Delta \mathbf{z}_i \mid \mathbf{x}_i \in \mathcal{D}_{\text{tr}}, y_i = 1, c_i \in \mathcal{C}_{\text{known}} \,\} \tag{20}$$

Here, $y_i = 1$ indicates that a sample is an attack, and $c_i \in \mathcal{C}_{\text{known}}$ indicates that its specific attack class belongs to the set of known attack classes defined in Section 3.2. Therefore, $\mathcal{A}^{\Delta}_{\text{known}}$ contains only contribution-deviation representations from known attacks. The held-out attacks are not involved in its construction.

Different known attacks may be driven by different combinations of features. Their contribution deviations may therefore exhibit a multimodal structure. A single global center is not sufficient to represent these different patterns. To model this structure more compactly, we learn M Deviation Semantic Prototypes (DSPs) from $\mathcal{A}^{\Delta}_{\text{known}}$. Let the set of all contribution-deviation prototypes be $\mathcal{P} = \{p_m\}_{m=1}^{M}$, where M denotes the number of prototypes and $p_m \in \mathbb{R}^k$ denotes the m-th representative contribution-deviation prototype. Each prototype represents a typical attack contribution-deviation pattern. It does not represent an attack sample center in the raw traffic feature space. The prototypes are learned by minimizing the squared Euclidean distance between each known attack DeltaSHAP vector and its nearest prototype.

$$\{\boldsymbol{p}_m\}_{m=1}^{M} = \arg\min_{\{\boldsymbol{p}_m\}_{m=1}^{M}} \sum_{\Delta \mathrm{z}_i \in \mathcal{A}_{\text{known}}{}^{\Delta}} \min_{1 \le m \le M} \|\Delta \mathrm{z}_i - \boldsymbol{p}_m\|_2{}^2 \tag{21}$$

The term $\|\Delta z_i - p_m\|_2^2$ measures the squared Euclidean distance between the dimension-wise contribution deviation of a known attack sample and the m-th prototype. This objective allows different contribution-deviation structures to be represented by different prototypes. A limited number of prototypes can therefore provide a compact representation of the multimodal deviation patterns observed in known attacks during training.

For any sample x_i to be detected, its DeltaSHAP representation Δz_i is first calculated using Eq. (19). We then measure its similarity to the representative deviation patterns learned from known attacks. The Euclidean distance to the nearest contribution-deviation prototype is denoted by $d_{\text{pat}}(x_i)$ and is defined in Eq. (22).

$$d_{\text{pat}}(\boldsymbol{x}_i) = \min_{1 \le m \le M} \|\Delta \text{z}_i - \boldsymbol{p}_m\|_2 \tag{22}$$

A smaller $d_{\text{pat}}(x_i)$ indicates that the structure of dimension-wise contribution deviations in the sample is more similar to a representative attack pattern observed during training.

The index of the nearest contribution-deviation prototype is denoted by m_i^* and is further defined in Eq. (23).

$$m_i^* = \arg\min_{1 \le m \le M} \|\Delta \mathbf{z}_i - \mathbf{p}_m\|_2 \tag{23}$$

The nearest-prototype index m_i^* identifies which learned attack deviation pattern is most similar to the current sample. It also preserves useful structural information for later explanation.

To convert the nearest-prototype distance into a similarity score that increases with pattern similarity, we define the contribution-deviation pattern similarity $s_{\text{pat}}(x_i)$using an exponential decay function.

$$s_{\text{pat}}(\boldsymbol{x}_i) = \exp\left(-\frac{d_{\text{pat}}(\boldsymbol{x}_i)}{\lambda + \varepsilon}\right) \tag{24}$$

The term λ is the decay scale for the nearest-prototype distance, and ε is a small positive constant used to avoid division by zero. A smaller distance produces a larger $s_{\text{pat}}(x_i)$. This indicates that the current contribution-deviation structure is closer to a representative attack pattern observed during training.

The decay scale is determined from the prototype-distance distribution of known attacks available during training. We first construct the set of nearest-prototype distances for all known attack samples, denoted by $\mathcal{D}_{\text{known}}^{\text{pat}}$.

$$\mathcal{D}_{\text{known}}{}^{\text{pat}} = \left\{ d_{\text{pat}}(\boldsymbol{x}_i) | \Delta \text{z}_i \in \mathcal{A}_{\text{known}}{}^{\Delta} \right\} \tag{25}$$

The distribution $\mathcal{D}_{\text{known}}^{\text{pat}}$ describes the distances between known attacks and their nearest learned contribution-deviation prototypes. Its median is denoted by $\tilde{d}_{\text{known}} = median(D_{\text{known}}^{\text{pat}})$. The decay scale λ is then determined according to Eq. (26).

$$\lambda = \begin{cases} \tilde{d}_{\text{known}}, \tilde{d}_{\text{known}} \geq \varepsilon, \\ \text{mean}(\mathcal{D}_{\text{known}}^{\text{pat}}) + \varepsilon, \tilde{d}_{\text{known}} < \varepsilon. \end{cases} \tag{26}$$

Here, $\varepsilon = 10^{-8}$ is a fixed numerical stability constant used for near-zero scale checking and to avoid division by zero in degenerate cases. When the median nearest-prototype distance is smaller than this numerical tolerance, it can no longer provide a reliable decay scale. In this case, the mean distance is used to describe the overall scale of the remaining nonzero distances. A small ε is also added to ensure a positive scale.

Different datasets may produce different similarity ranges and prototype-distance scales. We therefore apply robust scaling to the pattern similarity $s_{\text{pat}}(x_i)$ before evidence fusion. The scaling reference is constructed from the pattern similarities of normal training samples, denoted by $\mathcal{S}_N^{\text{pat}}$.

$$\mathcal{S}_N^{\text{pat}} = \{ s_{(\text{pat})(\mathbf{x}_i)} \mid \mathbf{z}_i \in \mathcal{Z}_N^{\text{tr}} \} \tag{27}$$

The set $\mathcal{S}_N^{\text{pat}}$ describes the similarity between normal training samples and the representative deviation patterns learned from known attacks. It serves as the normal reference for scaling the pattern evidence.

The robustly scaled contribution-deviation pattern evidence is denoted by $E_{\text{pat}}(x_i)$ and is then defined in Eq. (28).

$$E_{\text{pat}}(\boldsymbol{x}_i) = \frac{s_{\text{pat}}(\boldsymbol{x}_i) - \text{median}(\mathcal{S}_N{}^{\text{pat}})}{\text{IQR}(\mathcal{S}_N{}^{\text{pat}}) + \varepsilon} \tag{28}$$

The scaling uses $\text{median}(\mathcal{S}_N^{\text{pat}})$ and $\text{IQR}(\mathcal{S}_N^{\text{pat}})$, which are the median and interquartile range of the pattern similarities obtained from normal training samples, respectively. A larger $E_{\text{pat}}(x_i)$ indicates that the dimension-wise contribution deviation of the current sample is more similar to a representative attack deviation structure observed during training.

The resulting contribution-deviation pattern evidence $E_{\text{pat}}(x_i)$ measures how closely the current sample matches representative deviation structures of attacks. The nearest prototype m_i^* provides information about the most similar learned deviation pattern. In addition, DeltaSHAP dimensions with large absolute values identify the main contribution changes within that pattern.

3.6 AE-DSP Fusion Detection and Explainable Output

After modeling and scaling, the overall deviation evidence $E_{\mathrm{dev}}(x_i)$ from Section 3.4 and the contribution-deviation pattern evidence $E_{\mathrm{pat}}(x_i)$ from Section 3.5 can be jointly used for anomaly detection in the unified contribution space. We combine the two types of evidence using a weighted fusion strategy. The final fusion score of sample x_i is denoted by $S_{\mathrm{fus}}(x_i)$ and is defined in Eq. (29).

$$S_{\mathrm{fus}}(\boldsymbol{x}_i) = \alpha E_{\mathrm{dev}}(\boldsymbol{x}_i) + \gamma E_{\mathrm{pat}}(\boldsymbol{x}_i) \tag{29}$$

A larger fusion score indicates stronger combined attack evidence.

The threshold is calibrated only on the validation set and remains fixed during testing; held-out attacks are not used for threshold selection.

Given the fusion score $S_{\mathrm{fus}}(x_i)$ and the fixed detection threshold τ^*, let $\hat{y}_i$ denote the final binary prediction of sample x_i. It is defined in Eq. (30).

$$\hat{y}_i = \begin{cases} 1, & S_{\mathrm{fus}}(\boldsymbol{x}_i) > \tau^* \\ 0, & S_{\mathrm{fus}}(\boldsymbol{x}_i) \le \tau^* \end{cases} \tag{30}$$

A prediction of $\hat{y}_i = 1$ means that the sample is classified as an attack, while $\hat{y}_i = 0$ means that it is classified as normal. Both known attacks and held-out attacks are detected using the same binary decision rule. The model does not need to know the specific class of an unseen attack in advance.

The nearest DSP identifies the representative attack deviation structure most similar to the sample, while the dominant absolute DeltaSHAP dimensions locate the main contribution changes. Together with the overall and pattern evidence, these quantities provide instance-level explanatory evidence for the final decision.

4 Experimental Evaluation

4.1 Datasets and Held-Out Attack Settings

To evaluate the detection capability of AE-DSP under different network traffic distributions and attack scenarios, we conduct experiments on three public network intrusion detection datasets, including NSL-KDD, UNSW-NB15, and CICIDS2017. These datasets differ substantially in sample size, feature composition, and attack types. They therefore provide diverse settings for evaluating the model ability to generalize to attacks that are unseen during training.

Held-out attack experiments follow the class structure of each dataset. NSL-KDD uses the standard KDDTrain+ and KDDTest+ split, with attacks absent from training but present in testing treated as unseen; the experiment is repeated with seeds 42, 43, and 44. UNSW-NB15

uses five held-out classes—Analysis, Backdoor, DoS, Reconnaissance, and Shellcode—and CICIDS2017 uses nine—Bot, DDoS, DoS_GoldenEye, DoS_Hulk, DoS_Slowhttptest, DoS_slowloris, FTP_Patator, PortScan, and SSH_Patator. For each task, the target class is removed from training and validation and introduced only during testing, while all remaining attack classes are retained as known attacks.

4.2 Experimental Protocol, Implementation Details, and Evaluation Metrics

All experiments follow the held-out protocol defined in Section 3.2; preprocessing, model fitting, contribution-space construction, parameter selection, and threshold calibration use only training and validation data without access to the target held-out attacks.

AE-DSP performs binary normal/attack detection; held-out class labels are used only to construct and evaluate the held-out tasks.

4.2.1 Experimental Details and Parameter Settings

Dataset-specific splitting and preprocessing are applied according to the original data organization, while all comparison methods use identical train/validation/test indices for the same held-out task. Table 1 summarizes the resulting data construction, preprocessing, and contribution-space dimensions.

Table 1. Data Construction, Preprocessing, and Final Contribution Dimensions for the Three Datasets

Dataset	Data Split and Sample Size	Preprocessing	SHAP Contribution Subspace Dimension
NSL-KDD	KDDTrain+ is stratified by the normal or attack label. 75% is used for training and 25% for validation. The training and validation sets contain 94,479 and 31,494 samples, respectively. The complete KDDTest+ set with 22,544 samples is used for testing.	One-hot encoding is applied to protocol_type, service, and flag. The remaining numerical features are standardized.	50
UNSW-NB15	For each held-out task, the target attack is completely removed from the official training set. The remaining data are stratified by the normal or attack label, with 75% used for training and 25% for validation. The official testing set is used as the test set and retains the target held-out attack. The actual sample size varies slightly across held-out classes.	One-hot encoding is applied to proto, service, and state. The remaining numerical features are standardized.	50

Dataset	Data Split and Sample Size	Preprocessing	SHAP Contribution Subspace Dimension
CICIDS2017	After removing the held-out attack, the available training-stage data are divided into approximately 64% training, 16% validation, and 20% known-test data. The test set consists of the known-test samples and all held-out attack samples. To control computational cost, the numbers of samples in the training, validation, and test sets are capped at 120,000, 40,000, and 80,000, respectively.	Identifier and environment-related fields are removed. Infinite and missing values are handled, and numerical features are standardized.	80

All preprocessing parameters are estimated from the corresponding training data and then fixed for validation and testing. LightGBM and TreeSHAP are used as the base detector and contribution estimator, respectively. The final Global Top-k dimensions are 50 for NSL-KDD and UNSW-NB15 and 80 for CICIDS2017; all normal-reference statistics are estimated only from normal training samples. Other key parameter settings and sensitivity results are reported in Section 4.6.

The experiments are conducted on Ubuntu 22.04 using Python 3.10, LightGBM 4.6.0, SHAP 0.49.1, Scikit-learn 1.7.2, and PyTorch 2.12.0. The server is equipped with an Intel Xeon Gold 6530 processor and one NVIDIA RTX 5090 GPU.

4.2.2 Evaluation Metrics

We report Accuracy, Precision, Recall, and F1-score as standard binary classification metrics. To separately evaluate detection performance on attacks seen and unseen during training, we further use Known Recall and New Recall.

$$R_{\text{known}} = \frac{TP_{\text{known}}}{TP_{\text{known}} + FN_{\text{known}}} \tag{31}$$

$$R_{\text{new}} = \frac{TP_{\text{new}}}{TP_{\text{new}} + FN_{\text{new}}} \tag{32}$$

Known Recall evaluates detection on attacks observed during training, whereas New Recall evaluates the held-out attack classes. F1-score is used as the main metric for overall detection performance, while New Recall evaluates generalization to zero-day attacks.

4.3 Overall Performance Comparison

Table 2 presents the main results of AE-DSP on the three datasets. Accuracy, Precision, Recall, F1-score, Known Recall, and New Recall are reported for all datasets. For NSL-KDD, the reported values are the averages of three independent runs with random seeds 42, 43, and

44. For UNSW-NB15 and CICIDS2017, the reported values are the arithmetic means over five and nine class-wise held-out tasks, respectively.

Table 2. Main Results on the Three Datasets Using Six Evaluation Metrics

Dataset	Accuracy	Precision	Recall	F1-score	Known Recall	New Recall
NSL-KDD	0.9226	0.9057	0.9643	0.9341	0.9589	0.9774
UNSW-NB15	0.9153	0.9021	0.9493	0.9250	0.9511	0.9350
CICIDS2017	0.9455	0.8539	0.9672	0.9025	0.9953	0.9251

AE-DSP achieves New Recall values of 0.9774, 0.9350, and 0.9251 on NSL-KDD, UNSW-NB15, and CICIDS2017, respectively, while maintaining Known Recall of 0.9589, 0.9511, and 0.9953. The corresponding F1-scores are 0.9341, 0.9250, and 0.9025, indicating consistent overall performance across the three held-out settings.

To comprehensively evaluate AE-DSP across different zero-day attack detection paradigms, we select representative methods covering generative data augmentation, deep reinforcement learning, active learning, representation transformation, open-set recognition, incremental learning, and zero-shot learning, including SA-JS-WGAN-GP (Mu et al. 2026), DRL-DQN (Hussain et al. 2026), ADQN (Wu et al. 2024), SKT-IDS (Zha et al. 2024), OSRDW (Shin et al. 2023), DenyZero (Yao et al. 2025), InfoGAN + OpenMax (Fang and Xie 2025), AIS-NIDS (Farrukh et al. 2024), and Zero-Shot MLP (Sarhan et al. 2023). All methods are evaluated using the same data splits, held-out settings, and evaluation protocol as AE-DSP.

Table 3. Performance Comparison on Three Datasets Under the Unified Held-Out Attack Protocol

Method	Accuracy	Precision	Recall	F1-score	Known Recall	New Recall
NSL-KDD						
SA-JS-WGAN-GP (Mu et al. 2026)	0.8367	0.8554	0.8578	0.8566	0.8779	0.8091
DRL-DQN (Hussain et al. 2026)	0.8866	0.8892	0.9145	0.9017	0.9403	0.8520
ADQN (Wu et al. 2024)	0.9185	0.9045	0.9578	0.9304	0.9553	0.9640
SKT-IDS (Zha et al. 2024)	0.8697	0.9035	0.8630	0.8828	0.9508	0.6501
OSRDW (Shin et al. 2023)	0.6806	0.6659	0.8797	0.7580	0.9230	0.7746
AE-DSP (Ours)	**0.9226**	**0.9057**	**0.9643**	**0.9341**	**0.9589**	**0.9774**
UNSW-NB15						
DenyZero (Yao et al. 2025)	0.9063	0.9010	0.9423	0.9212	0.9479	0.8552
AIS-NIDS (Farrukh et al. 2024)	0.9047	0.8553	0.8897	0.8423	0.8706	0.8995
OSRDW (Shin et al. 2023)	0.8997	0.8135	0.8954	0.8525	0.9183	0.8593
Zero-Shot MLP (Sarhan et al. 2023)	0.9096	0.8959	0.9108	0.9033	0.9497	0.8554
AE-DSP (Ours)	**0.9153**	**0.9021**	**0.9493**	**0.9250**	**0.9511**	**0.9350**

Method	Accuracy	Precision	Recall	F1-score	Known Recall	New Recall
CICIDS2017						
DRL-DQN (Hussain et al. 2026)	0.9038	0.8392	0.9034	0.8701	0.9482	0.8341
InfoGAN + OpenMax (Fang and Xie 2025)	0.9330	0.8941	0.9427	0.9178	0.9577	0.9220
DenyZero (Yao et al. 2025)	0.9439	0.9032	0.9459	0.9241	0.9771	0.9166
AIS-NIDS (Farrukh et al. 2024)	0.8364	0.8677	0.8893	0.8584	0.9038	0.8724
AE-DSP (Ours)	**0.9455**	**0.8539**	**0.9672**	**0.9025**	**0.9953**	**0.9251**

As shown in Table 3, AE-DSP achieves the highest New Recall among the compared methods on all three datasets, reaching 0.9774 on NSL-KDD, 0.9350 on UNSW-NB15, and 0.9251 on CICIDS2017. On NSL-KDD, this is 1.34 percentage points higher than ADQN, while on UNSW-NB15 it is 3.55 percentage points higher than AIS-NIDS. AE-DSP also maintains Known Recall values of 0.9589, 0.9511, and 0.9953 and F1-scores of 0.9341, 0.9250, and 0.9025 on the three datasets, respectively.

On CICIDS2017, although AE-DSP achieves the highest New Recall and Known Recall, its Precision of 0.8539 is lower than the 0.9032 achieved by DenyZero, resulting in a lower F1-score of 0.9025 compared with 0.9241 for DenyZero. This indicates that the main advantage of AE-DSP on CICIDS2017 lies in its coverage of unseen attacks and its ability to preserve known attack detection performance.

Across the three datasets, the advantage of AE-DSP in New Recall shows that task-relevant anomaly evidence built from discriminative knowledge available during training can improve the coverage of unseen attacks. SHAP contribution representations transform raw traffic features into sample-level decision evidence that is directly related to the detector's malicious prediction. This allows the abnormality of unseen attacks to be modeled in a task-relevant contribution space. At the same time, the overall deviation from normal contribution behavior provides class-independent anomaly evidence without using information from the target attack class. The structure of dimension-wise contribution deviations further preserves the deviation direction, relative strength, and composition of different contribution changes. The following ablation experiments further evaluate the contribution of task-relevant SHAP representations and the joint modeling of these two types of anomaly evidence to the improvement in New Recall.

4.4 Class-Wise Held-Out Performance and Difficult-Class Analysis

Different attack classes vary substantially in traffic behavior, sample size, and contribution patterns. Reporting only the overall results may therefore hide performance

differences across specific held-out attacks. Section 4.4 presents the five class-wise held-out results on UNSW-NB15 and the class-wise results on CICIDS2017. The analysis focuses mainly on New Recall for different held-out attacks. Known Recall, Precision, and F1-score are also considered to examine unseen attack detection, preservation of known attack detection, and the difficulty of different attack types.

4.4.1 Class-Wise Held-Out Performance

To examine performance variations across unseen attack classes, Table 4 reports the class-wise held-out results on UNSW-NB15 and CICIDS2017. In each task, only the target attack class is excluded from training and validation, while the remaining attack classes are retained as known attacks.

Table 4. Class-Wise Held-Out Attack Performance on UNSW-NB15 and CICIDS2017

Held-out Attack	Accuracy	Precision	Recall	F1-score	Known Recall	New Recall
UNSW-NB15						
Analysis	0.9204	0.9024	0.9593	0.9299	0.9587	0.9985
Backdoor	0.9116	0.9003	0.9440	0.9216	0.9433	0.9966
DoS	0.9165	0.9008	0.9534	0.9263	0.9505	0.9819
Reconnaissance	0.9092	0.9049	0.9332	0.9188	0.9460	0.7800
Shellcode	0.9190	0.9020	0.9568	0.9286	0.9572	0.9180
CICIDS2017						
Bot	0.8990	0.6640	0.9986	0.7976	0.9999	0.9170
DDoS	0.9575	0.8848	0.9971	0.9376	0.9996	0.9954
DoS_GoldenEye	0.9943	0.9890	0.9836	0.9863	0.9889	0.9282
DoS_Hulk	0.8888	0.9002	0.8077	0.8515	0.9999	0.7534
DoS_Slowhttptest	0.9504	0.8141	0.9793	0.8891	0.9783	0.9999
DoS_slowloris	0.9952	0.9841	0.9923	0.9882	0.9920	0.9975
FTP_Patator	0.9553	0.8221	0.9990	0.9019	0.9997	0.9901
PortScan	0.9349	0.8674	0.9574	0.9102	0.9999	0.9360
SSH_Patator	0.9342	0.7596	0.9902	0.8597	0.9999	0.8087

On UNSW-NB15, Analysis, Backdoor, and DoS achieve New Recall values of 0.9985, 0.9966, and 0.9819, respectively, while Shellcode reaches 0.9180 and Reconnaissance decreases to 0.7800. Known Recall remains above 0.9400 across all five tasks. On CICIDS2017, DDoS, DoS_Slowhttptest, DoS_slowloris, and FTP_Patator achieve New Recall values close to or above 0.9900, whereas DoS_Hulk and SSH_Patator decrease to 0.7534 and 0.8087, respectively. Bot achieves a New Recall of 0.9170 but a lower Precision of 0.6640.

These results indicate that the difficulty of unseen attack detection varies substantially across attack classes.

4.4.2 Analysis of Difficult Held-Out Classes

Based on the class-wise results in Table 4, we further analyze four representative difficult held-out classes: Reconnaissance, Bot, DoS_Hulk, and SSH_Patator. Reconnaissance and DoS_Hulk mainly exhibit lower unseen-attack recall, Bot mainly suffers from low Precision, while SSH_Patator shows limitations in both metrics. Their high Known Recall values indicate that the main performance degradation is concentrated on the held-out classes.

From the perspective of contribution features and traffic behavior, these difficult classes show different forms of overlap with normal patterns. Reconnaissance is mainly associated with important contributions from TTL, destination port access frequency, connection state, and service access features. Its anomaly evidence is relatively dispersed. Some low-frequency scanning behavior may therefore resemble normal service access. The main contributions for Bot and DoS_Hulk are concentrated on TCP window and packet length related features. The stealthy and low-volume communication of Bot may produce contribution patterns similar to normal background connections. DoS_Hulk may contain more diverse traffic subpatterns within the same class. A single flow record of SSH_Patator may also appear similar to normal SSH access. Cross-connection behavior such as repeated login attempts and time intervals may not be fully captured by the existing features of an individual flow record.

Overall, the difficult cases of AE-DSP mainly fall into two categories. The first occurs when unseen attacks strongly overlap with normal traffic in the existing flow-level features and the task-relevant SHAP contribution space. In this case, the overall deviation from the normal contribution structure may not be sufficiently clear. The second occurs when an attack class contains substantial internal variation. In this case, a limited set of contribution-deviation prototypes learned from known attacks may not cover all unseen deviation patterns. These observations also suggest possible directions for future improvement. Longer-term behavioral information or richer contextual features could help improve the detection of stealthy attacks and attacks with strong internal heterogeneity.

4.5 Ablation Study

To evaluate the contributions of the task-relevant SHAP contribution space and the two complementary deviation measures, we compare four variants: Raw + AE, SHAP + AE-only, SHAP + DSP-only, and the complete SHAP + AE-DSP. Raw + AE models normal traffic directly in the raw feature space, whereas the other variants operate in the SHAP contribution space using overall deviation evidence, contribution-deviation pattern evidence, or their fusion, respectively. All variants follow the same held-out protocol and use the same

contribution dimensions within each dataset (k=50 for NSL-KDD and UNSW-NB15 and k=80 for CICIDS2017).

Table 5. Ablation Results on the Three Datasets Using Six Evaluation Metrics

Dataset	Method	Accuracy	Precision	Recall	F1-score	Known Recall	New Recall
NSL-KDD	Raw + AE	0.8739	0.9198	0.8528	0.8851	0.8770	0.7942
	SHAP Top-50 + AE-only	0.9184	0.9146	0.9448	0.9294	0.9618	0.9034
	SHAP Top-50 + DSP-only	0.7036	0.8560	0.5769	0.6891	0.7113	0.2513
	SHAP Top-50 + AE-DSP	0.9226	0.9057	0.9643	0.9341	0.9589	0.9774
UNSW-NB15	Raw + AE	0.8354	0.8829	0.8087	0.8440	0.8160	0.6396
	SHAP Top-50 + AE-only	0.8974	0.8909	0.9273	0.9086	0.9341	0.8002
	SHAP Top-50 + DSP-only	0.7302	0.7611	0.7433	0.7518	0.7439	0.7044
	SHAP Top-50 + AE-DSP	0.9153	0.9021	0.9493	0.9250	0.9511	0.9350
CICIDS2017	Raw + AE	0.8025	0.6161	0.5322	0.5685	0.5039	0.4964
	SHAP Top-80 + AE-only	0.9186	0.8203	0.9055	0.8541	0.9549	0.7590
	SHAP Top-80 + DSP-only	0.9202	0.7985	0.9355	0.8582	0.9892	0.7892
	SHAP Top-80 + AE-DSP	0.9455	0.8539	0.9672	0.9025	0.9953	0.9251

Compared with Raw + AE, SHAP + AE-only improves F1-score by 4.43, 6.46, and 28.56 percentage points on NSL-KDD, UNSW-NB15, and CICIDS2017, respectively, while New Recall increases by 10.92, 16.06, and 26.26 percentage points. These improvements show that SHAP contribution representations provide a more task-relevant basis for unseen-attack anomaly modeling than raw traffic attributes.

AE-only and DSP-only capture complementary anomaly evidence in the contribution space. AE-only performs better on NSL-KDD and UNSW-NB15, whereas DSP-only slightly outperforms AE-only on CICIDS2017, indicating different sensitivities across data distributions. Their fusion achieves the highest F1-score and New Recall on all three datasets. Compared with AE-only, the complete AE-DSP improves New Recall by 7.40, 13.48, and 16.61 percentage points on NSL-KDD, UNSW-NB15, and CICIDS2017, respectively. These results support the complementarity of overall deviation and dimension-wise contribution-deviation pattern evidence.

4.6 Random-Seed Stability and Parameter Sensitivity

To evaluate the reproducibility of AE-DSP, we first repeat the experiments on NSL-KDD using three random seeds. We then examine the sensitivity of detection performance to several key parameters around the final configuration. These parameters include the Top-k dimension

k, the number of DSP prototypes M, and the fusion weight γ for the contribution-deviation pattern evidence. Representative parameter variations are also evaluated on UNSW-NB15 and CICIDS2017 to examine parameter stability across different datasets.

4.6.1 Random-Seed Stability

Random-seed stability is evaluated on NSL-KDD using seeds 42, 43, and 44 under the same final configuration, with all settings unchanged except for random initialization.

Table 6. Stability Results on NSL-KDD Using Three Random Seeds

Seed	Accuracy	Precision	Recall	F1-score	Known Recall	New Recall
42	0.9209	0.9075	0.9588	0.9324	0.9447	0.9928
43	0.9293	0.9051	0.9784	0.9403	0.9731	0.9912
44	0.9175	0.9046	0.9558	0.9295	0.9589	0.9483
Mean ± Std	**0.9226 ± 0.0061**	**0.9057 ± 0.0015**	**0.9643 ± 0.0123**	**0.9341 ± 0.0056**	**0.9589 ± 0.0142**	**0.9774 ± 0.0253**

Across the three runs, F1-score is 0.9341 ± 0.0056 and New Recall is 0.9774 ± 0.0253. Accuracy and Precision also show low standard deviations of 0.0061 and 0.0015, respectively, while all three runs detect more than 94% of held-out attacks. These results indicate that the main performance is stable across different random initializations.

4.6.2 Parameter Sensitivity

We evaluate the sensitivity of AE-DSP to the Top-k dimension k, the number of DSP prototypes M, and the contribution-deviation pattern evidence weight γ. One parameter is varied at a time while the others remain fixed, and the final configuration is not reselected according to held-out test results. Table 7 reports the results on the three datasets.

Table 7. Parameter Sensitivity Results on the Three Datasets

Parameter	Value	F1-score	Known Recall	New Recall
Panel A. NSL-KDD				
Top-k dimension k	20	0.8997	0.9299	0.8035
	30	0.9243	0.9538	0.9037
	40	0.9325	0.9531	0.9733
	50	0.9324	0.9447	0.9928
	80	0.9393	0.9822	0.9400
Number of DSP prototypes M	8	0.9235	0.9368	0.9392
	12	0.9324	0.9447	0.9928
	16	0.9288	0.9352	0.9933
	20	0.9281	0.9330	0.9936
Contribution-deviation pattern evidence weight γ	0	0.9397	0.9832	0.9112
	0.8	0.9430	0.9810	0.9528

Parameter	Value	F1-score	Known Recall	New Recall
	1.2	0.9382	0.9696	0.9595
	1.8	0.9324	0.9447	0.9928
	2.2	0.9284	0.9363	0.9933
Panel B. UNSW-NB15				
Top-k dimension k	30	0.9225	0.9437	0.9220
	40	0.9250	0.9471	0.9282
	50	0.9248	0.9456	0.9283
	60	0.9246	0.9475	0.9396
	80	0.9233	0.9454	0.9337
Number of DSP prototypes M	8	0.9259	0.9476	0.9390
	12	0.9253	0.9473	0.9404
	16	0.9236	0.9453	0.9358
	20	0.9212	0.9397	0.9224
Contribution-deviation pattern evidence weight γ	1.2	0.9260	0.9477	0.9449
	1.5	0.9246	0.9470	0.9377
	1.8	0.9231	0.9442	0.9312
	2.1	0.9220	0.9428	0.9270
Panel C. CICIDS2017				
Top-k dimension k	60	0.8994	0.9977	0.9020
	80	0.8960	0.9963	0.9058
Number of DSP prototypes M	8	0.9055	0.9966	0.9170
	12	0.9062	0.9976	0.9007
Contribution-deviation pattern evidence weight γ	1	0.9040	0.9970	0.9010
	1.2	0.9037	0.9973	0.9137
	1.5	0.9058	0.9976	0.9129

On NSL-KDD, New Recall increases from 0.8035 at k=20 to 0.9928 at k=50 and then decreases to 0.9400 at k=80, indicating some sensitivity to the contribution-space dimensionality. Increasing M from 8 to 12 produces a clear improvement, whereas further increases result in only limited changes. As γ increases, New Recall generally improves while F1-score and Known Recall decrease, reflecting the balance between unseen-attack sensitivity and overall detection performance.

On UNSW-NB15 and CICIDS2017, F1-score and Known Recall vary only slightly across the evaluated settings, while New Recall shows relatively larger fluctuations. Overall, the results across the three datasets indicate that AE-DSP does not depend on a single narrowly defined parameter value within the evaluated ranges.

4.7 Quantitative Explainability Analysis

Quantitative explainability is evaluated from two perspectives: decision consistency of the identified DeltaSHAP dimensions and structural quality of the learned DSP prototypes. The

experiments use the final NSL-KDD configuration with random seed 42, and random masking results are averaged over 20 repetitions.

4.7.1 Masking-Based Validation of Contribution Importance

For each unseen attack sample, the k dimensions with the largest absolute DeltaSHAP values are replaced by the corresponding mean SHAP contributions of normal training samples, after which the AE-DSP evidence and prediction are recalculated. As a control, the same number of dimensions are randomly selected from the same Top-50 contribution subspace and masked using the same procedure. The unmasked New Recall is 0.9928.

Table 8. Results of Top-|DeltaSHAP| Masking and Random Masking

Masked dimensions k	Original New Recall	Top-|DeltaSHAP| masked New Recall	Top masking drop (percentage points)	Random masked New Recall
1	0.9928	0.9451	4.77	0.9919 ± 0.0006
3	0.9928	0.8677	12.51	0.9900 ± 0.0012
5	0.9928	0.8725	12.03	0.9889 ± 0.0018
10	0.9928	0.7808	21.20	0.9831 ± 0.0019

Masking the key DeltaSHAP dimensions causes substantially larger reductions in New Recall than random masking. For k=1, 3, and 10, the decreases are 4.77, 12.51, and 21.20 percentage points, compared with only 0.09, 0.28, and 0.97 percentage points under random masking. Although the effect is not strictly monotonic, it remains consistently stronger than random masking, supporting the decision consistency of the instance-level explanations.

4.7.2 Quality Analysis of DSP Contribution-Deviation Prototypes

We evaluate DSP prototype structure using prototype purity, normalized entropy, and cosine consistency, which measure attack-class concentration, class mixing, and directional consistency of the assigned DeltaSHAP vectors, respectively.

Prototype quality is evaluated using 43,972 known-attack samples from 21 attack classes assigned to 12 learned DSP prototypes. This analysis evaluates the internal structure of the learned prototypes rather than attack-classification performance on an independent test set.

Table 9. Overall Quality of DSP Contribution-Deviation Prototypes

Metric	Macro average	Weighted average	Interpretation
Prototype purity	0.8968	0.8966	Higher is better
Normalized entropy	0.1018	0.0991	Lower is better
Cosine consistency	0.9547	0.9604	Higher is better

The prototypes achieve macro/weighted purity values of 0.8968/0.8966, normalized entropy values of 0.1018/0.0991, and cosine consistency values of 0.9547/0.9604; 9 of the 12

prototypes have purity of at least 0.90. The combination of high purity, low entropy, and high cosine consistency indicates relatively concentrated attack-class compositions and consistent contribution-deviation directions, supporting the conclusion that the DSPs capture structured patterns rather than unstructured aggregations.

4.8 Computational Cost and Deployment Analysis

Computational cost is evaluated using the trained NSL-KDD model with random seed 42 on one NVIDIA RTX 5090 GPU. The measurement covers preprocessing, TreeSHAP contribution generation, post-SHAP AE-DSP inference, end-to-end latency, throughput, and memory usage, excluding model training time. Batch sizes of 1, 32, and 128 are used to evaluate single-sample and micro-batch inference.

Table 10 reports the end-to-end inference results and resource usage. P50, P95, and P99 denote the median, 95th-percentile, and 99th-percentile latency, respectively; mean latency per sample and throughput characterize per-sample processing efficiency.

Table 10. Computational Cost and Resource Usage of AE-DSP on NSL-KDD

Panel A. End-to-End Inference Performance

Batch size	P50 latency (ms/batch)	P95 latency (ms/batch)	P99 latency (ms/batch)	Mean latency (ms/sample)	Throughput (samples/s)
1	61.0327	67.3962	76.97	61.66	16.22
32	62.1273	77.2597	81.36	2	499.88
128	63.1054	70.2372	83.86	0.498	2009.66

Panel B. P50 Latency Breakdown

Batch size	CPU preprocessing (ms/batch)	GPU TreeSHAP (ms/batch)	GPU post-SHAP AE-DSP (ms/batch)	End-to-end (ms/batch)
1	2.48	57.76	0.6	61.03
32	2.5	58.2	0.64	62.13
128	2.62	59.02	0.64	63.11

Panel C. Memory Usage

Resource metric	Peak usage
CPU process RSS	Approximately 1475 MB
PyTorch peak allocated memory	32.83 MiB
PyTorch peak reserved memory	34.00 MiB
Entire GPU process peak memory	626.00 MiB

End-to-end results show that the average per-sample latency decreases from 61.66 ms at batch size 1 to 2.00 ms and 0.498 ms at batch sizes 32 and 128, while throughput increases from 16.22 to 499.88 and 2009.66 samples/s. Meanwhile, P50 batch latency increases only from 61.03 to 63.11 ms, showing that micro-batching substantially amortizes the fixed inference overhead.

Component-level analysis shows that GPU TreeSHAP requires 57.76–59.02 ms per batch and accounts for approximately 93.5%–94.7% of end-to-end P50 latency. In comparison, CPU preprocessing requires 2.48–2.62 ms and the post-SHAP AE-DSP stage only 0.60–0.64 ms. Therefore, SHAP contribution generation, rather than the proposed AE-DSP modeling itself, is the dominant inference bottleneck.

Peak CPU process RSS is approximately 1475 MB, while the entire GPU process reaches 626 MiB. The PyTorch allocator records only 32.83 MiB allocated and 34.00 MiB reserved because GPU TreeSHAP and other CUDA components are not fully tracked by it. Thus, the total GPU memory footprint remains below 1 GiB.

These results indicate that AE-DSP is feasible for GPU-supported flow-level detection with micro-batch processing, while the post-SHAP AE-DSP stage itself introduces only limited additional computational cost.

4.9 Visualization of the SHAP Contribution Space and DSP Prototypes

We select representative datasets to visualize the raw feature space, the SHAP contribution space, and the DSP contribution-deviation prototypes. These visualizations are used to examine the effect of the SHAP-space transformation and the deviation structures captured by the DSP prototypes. t-SNE is used to compare sample distributions before and after the space transformation. Prototype heatmaps are used to show the deviation patterns of different DSP contribution-deviation prototypes across DeltaSHAP dimensions.

4.9.1 Raw Feature Space versus SHAP Contribution Space

To visually examine how the sample structure changes after transformation from the raw traffic feature space to the SHAP contribution space, we select UNSW-NB15 as a representative dataset. t-SNE is applied to both the raw Top-k feature vectors and the Global Top-k SHAP contribution vectors. Different colors are used to represent normal traffic, known attacks, and held-out unseen attacks.

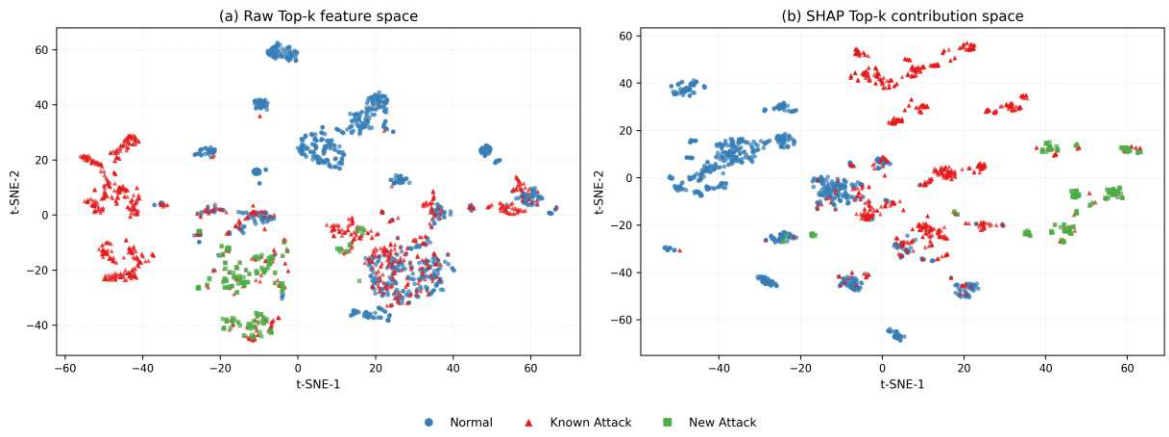


Fig. 2 t-SNE Comparison Between the Raw Feature Space and the SHAP Contribution Space on UNSW-NB15

As shown in Fig. 2, the raw feature space contains substantial overlap among different sample types, whereas the SHAP contribution space exhibits clearer local structures for normal traffic, known attacks, and held-out unseen attacks. This indicates that the transformation reorganizes samples according to task-relevant decision contributions rather than raw traffic similarity. UNSW-NB15 is used as a representative visualization, while cross-dataset quantitative evidence is provided by the ablation experiments in Section 4.5.

4.9.2 Visualization of DSP Contribution-Deviation Prototypes

We select NSL-KDD as the representative dataset to analyze the deviation structures learned by DSP from known attacks. This also avoids presenting similar heatmaps for all three datasets. The eight DSP contribution-deviation prototypes with the largest numbers of assigned samples are visualized. Each row of the heatmap corresponds to one DSP contribution-deviation prototype. Each column represents an important SHAP feature. The color indicates the DeltaSHAP center value of the corresponding prototype relative to the normal contribution baseline.

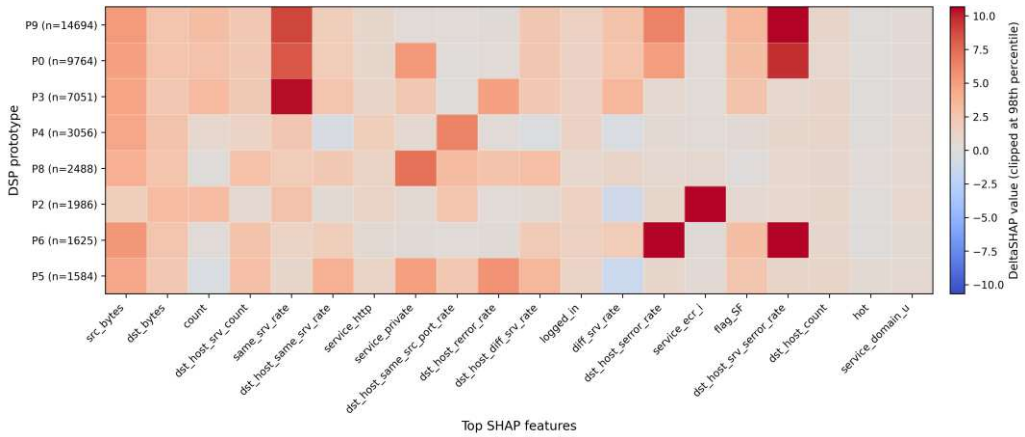


Fig. 3 Heatmap of the Top-8 DSP Contribution-Deviation Prototypes on NSL-KDD

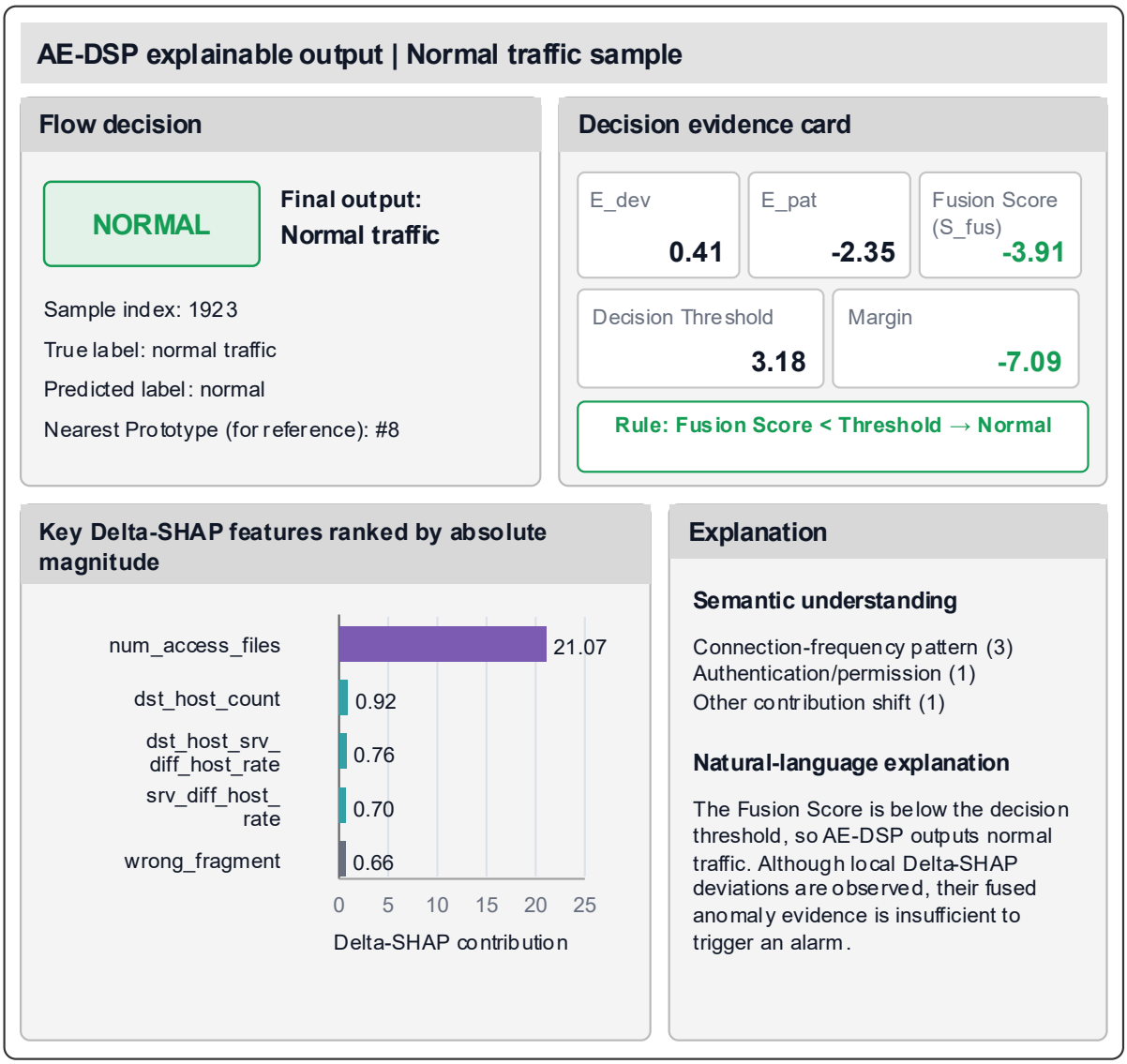
Fig. 3 shows that the learned DSP prototypes exhibit distinct combinations, directions, and strengths of contribution deviations rather than a single average attack pattern. The qualitative pattern in Fig. 3 is consistent with the quantitative prototype-quality results in Section 4.7.

4.10 Instance-Level Explanation Analysis

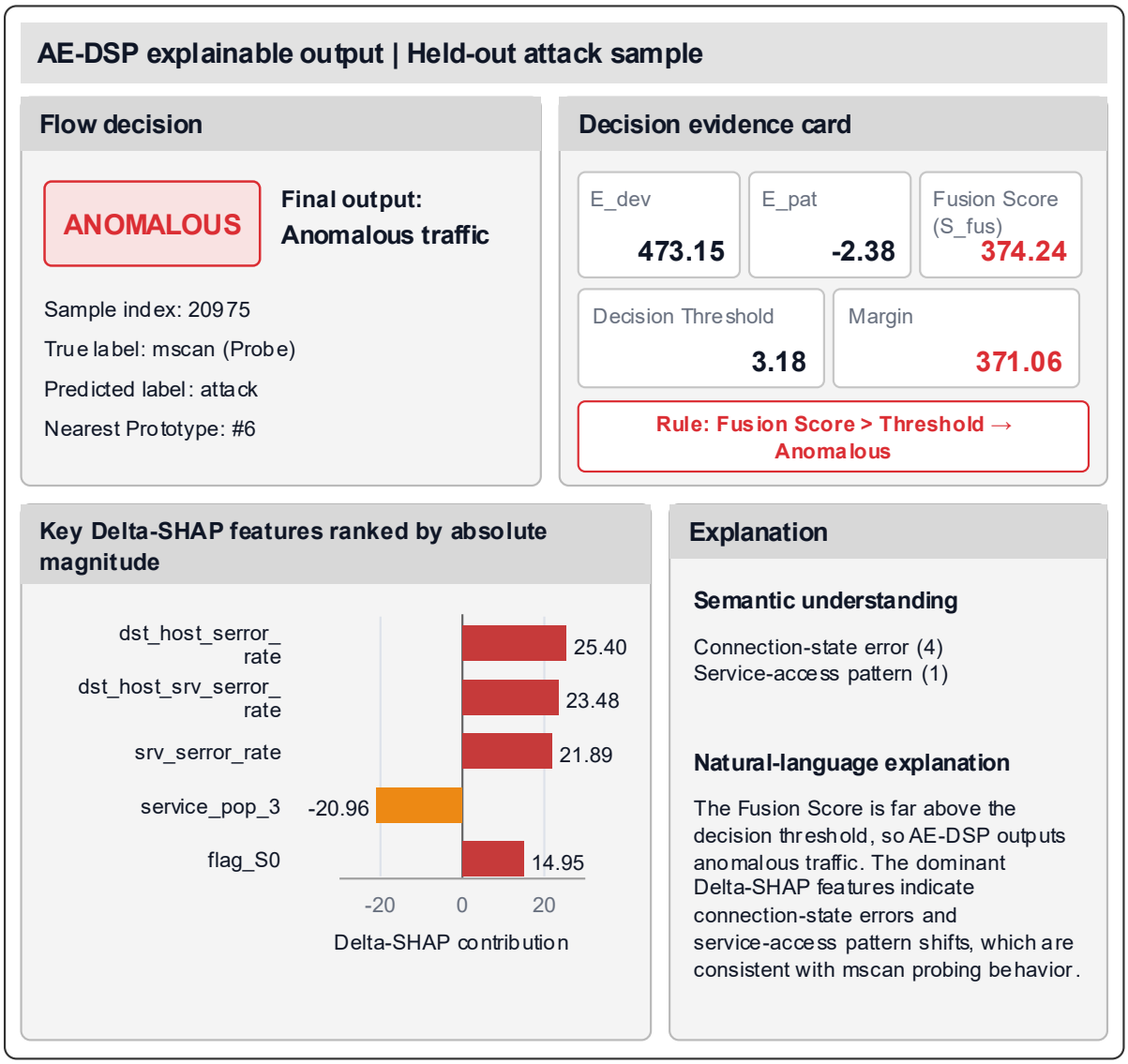
For each sample, AE-DSP provides the final label and Fusion Score together with the overall deviation evidence, contribution-deviation pattern evidence, nearest DSP prototype, and dominant DeltaSHAP dimensions. These outputs form an explanation path from the final detection result to the main sources of abnormal contribution.

4.10.1 Instance-Level Explanation of Normal and Unseen-Attack Samples

The sign of a DeltaSHAP value indicates the direction of deviation from the normal contribution baseline and does not alone determine the final label; the final decision is based on the fused overall and pattern evidence. To illustrate how the explanation information is organized, Fig. 4 presents one correctly identified normal sample and one correctly detected held-out mscan attack sample.



(a) Correctly identified normal traffic



(b) Correctly detected held-out mscan attack

Fig. 4 Complete Instance-Level Explainable Outputs for Normal Traffic and a Held-Out Unseen Attack

For the normal sample, the Fusion Score is below the detection threshold. Although some local DeltaSHAP deviations are present, they are not strong enough to form sufficient overall attack evidence. In contrast, the Fusion Score of the held-out mscan sample clearly exceeds the threshold. Its main contribution deviations are concentrated in dimensions related to connection-state errors and service-access patterns. This comparison shows that AE-DSP can jointly present the final decision, the two types of anomaly evidence, and the main sources of abnormal contribution, rather than providing only a single attack score.

For the held-out mscan sample, the dominant DeltaSHAP deviations involve connection-error rates, service-level error rates, TCP state, and service-access patterns. Their simultaneous deviations from the normal contribution baseline are consistent with the Fusion Score exceeding the detection threshold. Thus, DeltaSHAP characterizes the directions and

combinations of abnormal decision contributions rather than identifying only a single dominant feature.

4.10.2 Cross-Dataset Explanation Case

To examine the explanation approach across datasets, Fig. 5 presents one correctly detected held-out Analysis attack from UNSW-NB15 as a representative case.

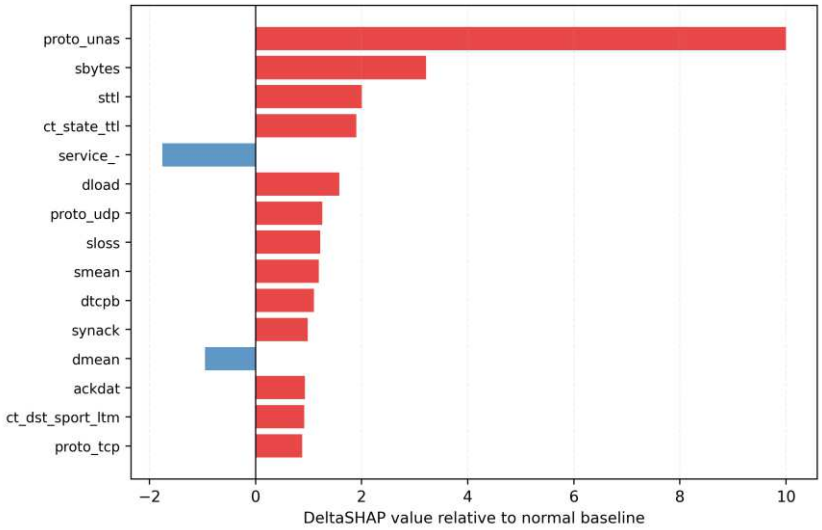


Fig. 5 DeltaSHAP Explanation of a Held-Out Analysis Attack on UNSW-NB15

The Analysis sample shows strong overall deviation evidence and contribution-deviation pattern evidence. Its main DeltaSHAP deviations involve several aspects, including protocol behavior, traffic volume, TTL, and connection state. Compared with the NSL-KDD mscan sample discussed above, the two unseen attacks rely on different raw features. Even so, both can be explained within the same contribution-space framework. Their detection results can be traced through the overall deviation evidence, contribution-deviation pattern evidence, and key DeltaSHAP dimensions.

A similar pattern is also observed for a DDoS instance in CICIDS2017. Its abnormal contributions are mainly associated with packet-length distributions, bidirectional traffic, connection active and idle times, and TCP flag behavior. Since the organization of its explanation is similar to the two examples above, we do not present an additional figure. These results show that the instance-level explanations of AE-DSP do not depend on a dataset-specific set of raw features. Instead, they are built on a unified SHAP contribution-deviation analysis framework.

5 Conclusions and Future Work

5.1 Conclusions

This paper proposes AE-DSP, an explainable zero-day intrusion detection framework that transforms sample-level SHAP contributions into a task-relevant contribution space. Within this space, AE-DSP jointly models overall deviation from normal decision evidence and dimension-wise contribution-deviation patterns, enabling unseen-attack detection and instance-level explanation within a unified representation.

Under the strict held-out protocol, AE-DSP achieves New Recall values of 0.9774, 0.9350, and 0.9251 and F1-scores of 0.9341, 0.9250, and 0.9025 on NSL-KDD, UNSW-NB15, and CICIDS2017, respectively. It achieves the highest New Recall in the unified comparisons, while ablation, stability, and quantitative explainability results support the effectiveness of the SHAP contribution representation, the complementarity of the two deviation evidences, and the decision relevance of the identified DeltaSHAP dimensions.

5.2 Limitations and Future Work

AE-DSP remains challenging when unseen attacks strongly overlap with normal traffic or exhibit substantial intra-class variation. Because the current framework relies mainly on flow-level features and contribution information generated by a fixed base detector, it may not fully capture long-term or contextual behavior. Future work can incorporate temporal, session-level, and host-level information and examine the transferability of the contribution space across different detectors and feature sets.

TreeSHAP contribution generation remains the main online computational bottleneck, whereas the post-SHAP AE-DSP stage is relatively lightweight. The current implementation is therefore more suitable for GPU-supported flow-level micro-batch detection, secondary analysis, and offline auditing than for packet-level line-rate blocking. Future work should investigate faster or approximate contribution computation and evaluate long-term stability in dynamic real-world networks.

References

Barnard P, Marchetti N, DaSilva LA (2022) Robust network intrusion detection through explainable artificial intelligence. IEEE Networking Letters, 4(3): 167–171. https://doi.org/10.1109/LNET.2022.3186589

Bendale A, Boult TE (2016) Towards open set deep networks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition: 1563–1572. https://doi.org/10.1109/CVPR.2016.173

Chen J, Li J, Qu X, Wang J, Wan J, Xiao J (2023) GAIA: Delving into Gradient-based Attribution Abnormality for Out-of-distribution Detection. Advances in Neural Information Processing Systems, 36: 79946–79958.

Fang J, Xie C (2025) Unknown intrusion traffic detection method based on unsupervised learning and open-set recognition. Scientific Reports, 15: article 17001. https://doi.org/10.1038/s41598-025-01084-1

Farrukh YA, Wali S, Khan I, Bastian ND (2024) AIS-NIDS: An intelligent and self-sustaining network intrusion detection system. Computers & Security, 144: article 103982. https://doi.org/10.1016/j.cose.2024.103982

Hindy H, Atkinson R, Tachtatzis C, Colin J-N, Bayne E, Bellekens X (2020) Utilising deep learning techniques for effective zero-day attack detection. Electronics, 9(10): article 1684. https://doi.org/10.3390/electronics9101684

Hussain S, Kalpana V, Raghavendran V, Gangatharan N, Mouleswararao B, Veeresha RK (2026) Machine learning-based zero-day attack detection using deep reinforcement learning. Information Security Journal: A Global Perspective. https://doi.org/10.1080/19393555.2026.2632663

Injadat M, Moubayed A, Nassif AB, Shami A (2021) Multi-stage optimized machine learning framework for network intrusion detection. IEEE Transactions on Network and Service Management, 18(2): 1803–1816. https://doi.org/10.1109/TNSM.2020.3014929

Le T-T-H, Kim H, Kang H, Kim H (2022) Classification and explanation for intrusion detection system based on ensemble trees and SHAP method. Sensors, 22(3): article 1154. https://doi.org/10.3390/s22031154

Lundberg SM, Lee S-I (2017) A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems, 30: 4765–4774.

Lundberg SM, Erion GG, Lee S-I (2018) Consistent individualized feature attribution for tree ensembles. arXiv preprint arXiv:1802.03888.

Mirsky Y, Doitshman T, Elovici Y, Shabtai A (2018) Kitsune: An ensemble of autoencoders for online network intrusion detection. In: Proceedings of Network and Distributed System Security Symposium (NDSS). https://doi.org/10.14722/ndss.2018.23204

Moustafa N, Slay J (2015) UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: Proceedings of 2015 Military Communications and Information Systems Conference (MilCIS): 1–6. https://doi.org/10.1109/MilCIS.2015.7348942

Mu Z, Shi X, Dogan S (2026) A novel solution for zero-day attack detection in IDS using self-attention and Jensen-Shannon divergence in WGAN-GP. Computer Networks, 282: article 112285. https://doi.org/10.1016/j.comnet.2026.112285

Pang G, Shen C, Cao L, van den Hengel A (2021) Deep learning for anomaly detection: A review. ACM Computing Surveys, 54(2): 1–38: article 38. https://doi.org/10.1145/3439950

Rivero J, Ribeiro B, Chen N, Silva Leite F (2017) A Grassmannian approach to zero-shot learning for network intrusion detection. In: Neural Information Processing: Proceedings of ICONIP 2017: 565–575. https://doi.org/10.1007/978-3-319-70087-8_59

Ruff L, Vandermeulen RA, Görnitz N, Deecke L, Siddiqui SA, Binder A, Müller E, Kloft M (2018) Deep one-class classification. In: Proceedings of 35th International Conference on Machine Learning, 80: 4393–4402.

Sarhan M, Layeghy S, Gallagher M, Portmann M (2023) From zero-shot machine learning to zero-day attack detection. International Journal of Information Security, 22: 947–959. https://doi.org/10.1007/s10207-023-00676-0

Sayem IM, Sayed MI, Saha S, Haque A (2024) ENIDS: A deep learning-based ensemble framework for network intrusion detection systems. IEEE Transactions on Network and Service Management, 21(5): 5809–5825. https://doi.org/10.1109/TNSM.2024.3414305

Sharafaldin I, Lashkari AH, Ghorbani AA (2018) Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: Proceedings of 4th International Conference on Information Systems Security and Privacy (ICISSP), 1: 108–116. https://doi.org/10.5220/0006639801080116

Shin G-Y, Kim D-W, Han M-M (2023) Open set recognition with dissimilarity weight for unknown attack detection. IEEE Access, 11: 102381–102390. https://doi.org/10.1109/ACCESS.2023.3277871

Soltani M, Ousat B, Jafari Siavoshani M, Jahangir AH (2023) An adaptable deep learning-based intrusion detection system to zero-day attacks. Journal of Information Security and Applications, 76: article 103516. https://doi.org/10.1016/j.jisa.2023.103516

Souza MMC, Pontes CT, Gondim JJC, Garcia LPF, DaSilva LA, Cavalcante EFM, Marotta MA (2025) A novel open set energy-based flow classifier for network intrusion detection. Computers & Security, 157: article 104569. https://doi.org/10.1016/j.cose.2025.104569

Tavallaee M, Bagheri E, Lu W, Ghorbani AA (2009) A detailed analysis of the KDD CUP 99 data set. In: Proceedings of 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA): 1–6. https://doi.org/10.1109/CISDA.2009.5356528

Wali S, Farrukh YA, Khan I (2025) Explainable AI and random forest based reliable intrusion detection system. Computers & Security, 157: article 104542. https://doi.org/10.1016/j.cose.2025.104542

Wei F, Li H, Zhao Z, Hu H (2023) xNIDS: Explaining deep learning-based network intrusion detection systems for active intrusion responses. In: Proceedings of 32nd USENIX Security Symposium: 4337–4354.

Wu Y, Hu Y, Wang J, Feng M, Dong A, Yang Y (2024) An active learning framework using deep Q-network for zero-day attack detection. Computers & Security, 139: article 103713. https://doi.org/10.1016/j.cose.2024.103713

Yao T, Wang H, Huang G, Wang Y, Yao J, Huang M (2025) Cost-aware and progressive input embedding-based rejection zero-day attack detection model. KSII Transactions on Internet and Information Systems, 19(10): 3625–3647. https://doi.org/10.3837/tiis.2025.10.016

Zha C, Wang Z, Fan Y, Zhang X, Bai B, Zhang Y, Shi S, Zhang R (2024) SKT-IDS: Unknown attack detection method based on Sigmoid Kernel Transformation and encoder–decoder architecture. Computers & Security, 146: article 104056. https://doi.org/10.1016/j.cose.2024.104056

Zhang J, Wang X, Jin Z, Zhu Z, Zhou J, Chen F, Chen H (2025) Splitting & Integrating: Out-of-Distribution Detection via Adversarial Gradient Attribution. In: Proceedings of the 42nd International Conference on Machine Learning, PMLR, 267: 76213–76224.

Statements and Declarations

Funding

Funding details are withheld from this blinded manuscript to preserve the anonymity of the authors. They are provided in the submission system and the separate author-information document.

Competing interests

The authors have no relevant financial or non-financial interests to disclose.

AI-assisted manuscript preparation

During the preparation of this work, the authors used ChatGPT (OpenAI) to assist with English translation, language polishing, and manuscript editing. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.