

## A Additional Details on GraphIDS and Baseline Models

**CBLOF [19].** CBLOF is an unsupervised anomaly detection method which identifies outliers based on a clustering algorithm. In this case we use K-means for clustering and we compute the anomaly scores based on the distance to the closest large cluster. As we do for other models, once the anomaly scores are computed, we search for the best threshold on the validation set and we apply it to the test set for the final classification.

**Anomal-E [2].** Anomal-E is the first model to apply self-supervised GNNs to network intrusion detection, demonstrating significant performance gains when applying anomaly detection methods to local graph representations compared to raw NetFlow features. The model uses an E-GraphSAGE encoder trained via a modified version of Deep Graph Infomax [22], and generates embeddings that are subsequently processed by traditional unsupervised anomaly detection algorithms, including PCA-based anomaly detection [23], Isolation Forest [24], CBLOF [19], and histogram-based outlier score [25]. For our evaluation, we retained the original GNN encoder configuration, which was already tuned for these datasets, as our attempts at further tuning did not yield performance gains. We did, however, explore and adjust the hyperparameters of the downstream anomaly detection components using the same ranges as in the original work, selecting the best-performing one for each dataset to be included in the main table.

During the preprocessing phase, the original authors report using target encoding for categorical features. Although the specific target used for encoding is not clarified in the paper, their public implementation<sup>3</sup> shows that attack labels are directly used as the target variable—introducing label leakage. This ground-truth information encoded in the input features allows downstream models to learn to identify attacks based on those statistics, undermining the validity of the unsupervised learning setting. To ensure a fair comparison and preserve the integrity of the evaluation, we removed the target encoding step in our implementation.

**SAFE [20].** SAFE is an anomaly detection framework that processes tabular network traffic data by first applying feature selection to discard irrelevant columns. The remaining features are then mapped into a 2D grid to create image-like embeddings, which a lightweight CNN-based masked autoencoder is trained to reconstruct, learning meaningful representations in the process. For novelty detection, the latent embeddings produced by the encoder are passed to a local outlier factor detector [26] to identify anomalies. In our evaluation, we tuned the hyperparameters of the LOF anomaly detection component. However, a brief exploration of alternative hyperparameters for the MAE module yielded no performance improvements. Given the high computational cost of tuning and the MAE’s limited discrimination ability, as observed in the original codebase, we concluded that further tuning would offer marginal gains and not meaningfully affect the conclusions.

In our experiments, we adopt different evaluation metrics, as the original implementation computes the F1-score for the normal class, effectively measuring the model’s ability to recognize benign traffic rather than attacks<sup>4</sup>. While this choice may be acceptable for balanced classes, it masks the model’s real performance on the highly imbalanced datasets we consider.

**T-MAE.** T-MAE refers to our Transformer-based masked autoencoder component, similar to the one used in GraphIDS but applied directly to raw NetFlow features. We use the same batching strategy (batch size of 64 with 512 flow embeddings per batch) and tune its learning rate, weight decay, and dropout. We found that a higher learning rate proved to be especially beneficial for the performance on the NF-UNSW-NB15-v3. However, despite this adjustment, T-MAE exhibits slower convergence, resulting in significantly longer training times. On average, it requires 2.21 hours per run, compared to just 0.87 hours for GraphIDS.

**SimpleAE.** The SimpleAE ablation replaces the Transformer with a fully connected autoencoder consisting of a two-layer MLP encoder and a two-layer MLP decoder with ReLU activations. It is trained end-to-end jointly with E-GraphSAGE on the same reconstruction objective, isolating the

---

<sup>3</sup><https://github.com/waimorris/Anomal-E/>, Apache License 2.0.

<sup>4</sup><https://github.com/ElvinLit/SAFE/>, No formal license available. Used with permission from the authors for research purposes only.

architectural benefit on top of our end-to-end reconstruction framework. To ensure a fair comparison, we explored different bottleneck dimensions and hyperparameters.

## B Extended Results and Comparative Analysis

### B.1 Qualitative Analysis of Detection Behavior

Figure 4 summarizes model performance across datasets through precision-recall curves. These plots illustrate that GraphIDS consistently matches or outperforms the baselines across a variety of settings.

To better understand GraphIDS’s behavior on specific attack types, we plot the distribution of anomaly scores (by density) for each dataset, as shown in Figures 5, 6, 7, and 8. To maintain clarity, we present representative examples without error bars. Each plot includes the classification threshold, allowing us to visualize which attack types were correctly detected and which ones were missed. In particular, GraphIDS’s lower performance on NF-UNSW-NB15-v2 is due to a higher rate of false positives, as also demonstrated by the t-SNE visualizations of the GNN and reconstructed embeddings in Figure 10. For the NF-CSE-CIC-IDS2018 datasets (both v2 and v3), the performance drop is primarily caused by misclassifications of Infiltration attacks. These involve delivering a malicious payload via email, which then attempts to exploit internal vulnerabilities by scanning the network. Because this behavior closely resembles normal traffic, GraphIDS struggles to reliably classify it as anomalous without prior knowledge of its specific signature, as illustrated in Figures 11 and 12. In contrast, the strong performance on NF-UNSW-NB15-v3 is reflected in the clear separation between benign and attack clusters in Figure 9.

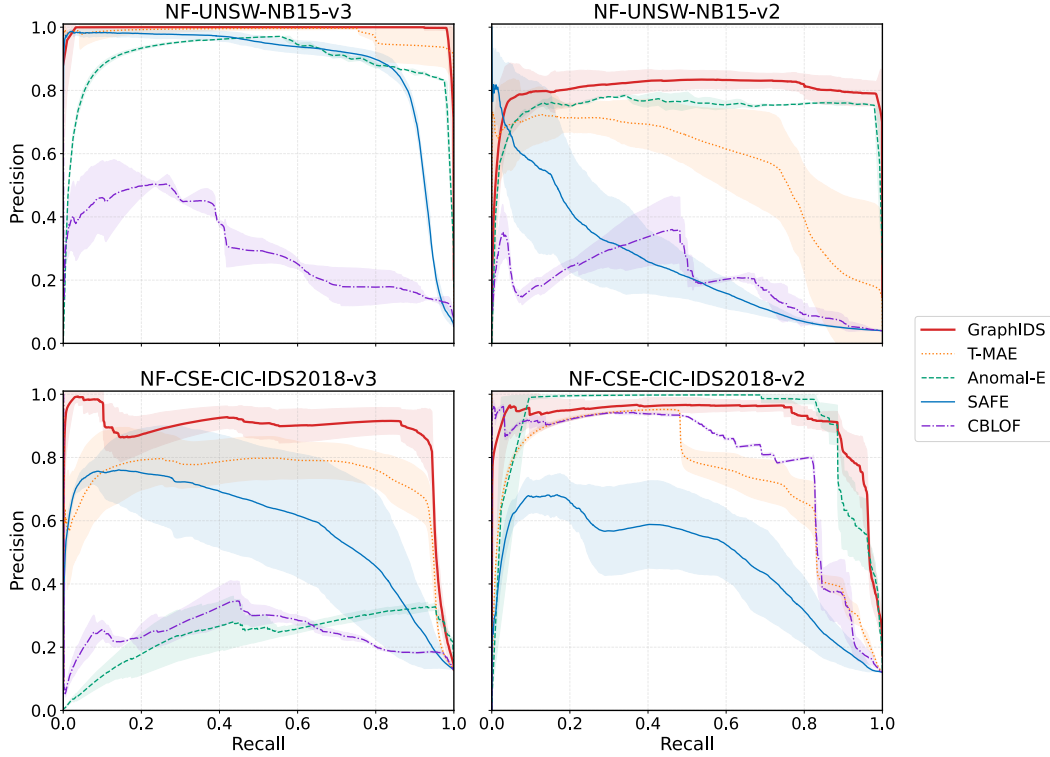


Figure 4: Precision-recall curves for all models on each dataset.

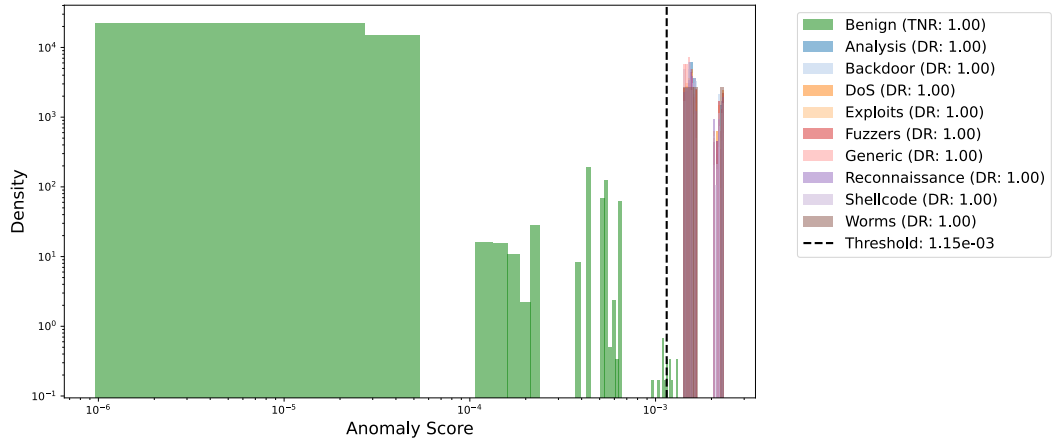


Figure 5: Anomaly score by attack type in NF-UNSW-NB15-v3.

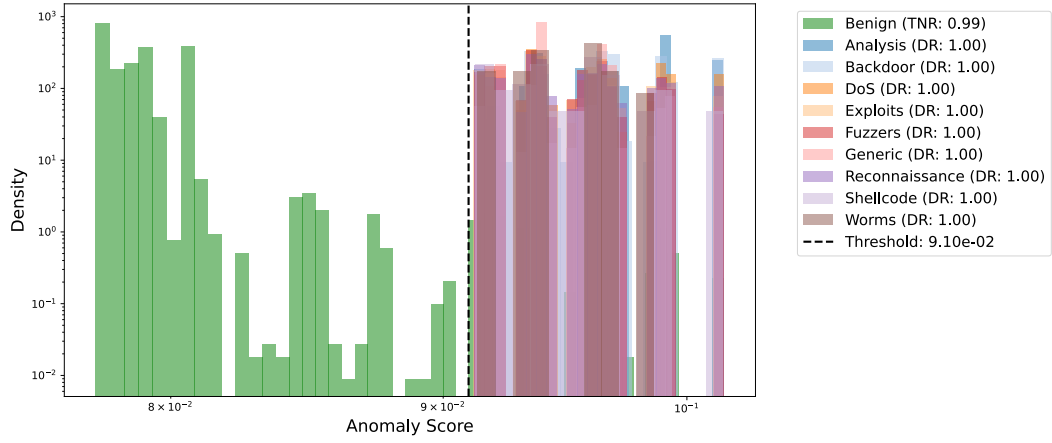


Figure 6: Anomaly score by attack type in NF-UNSW-NB15-v2.

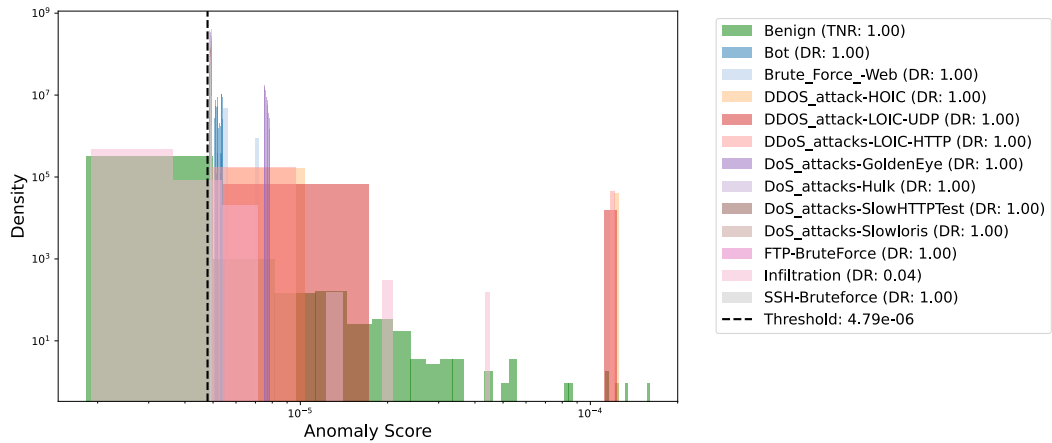


Figure 7: Anomaly score by attack type in NF-CSE-CIC-IDS2018-v3.

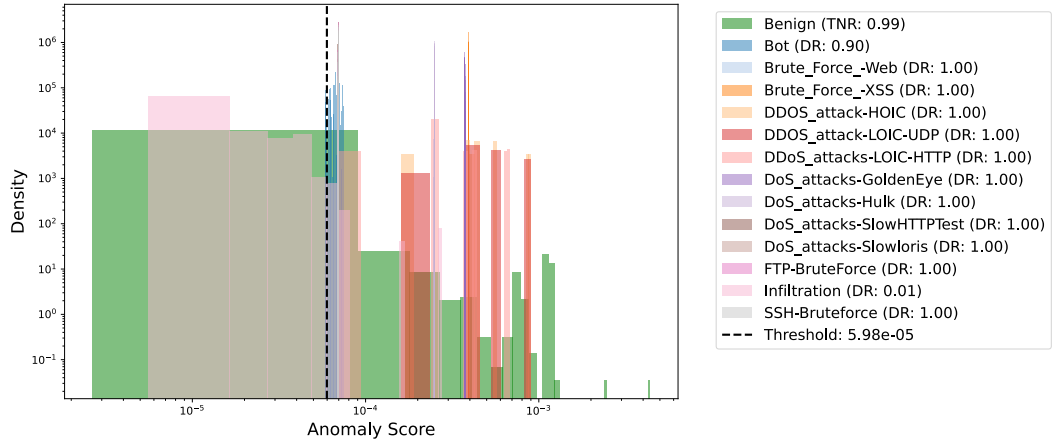


Figure 8: Anomaly score by attack type in NF-CSE-CIC-IDS2018-v2.

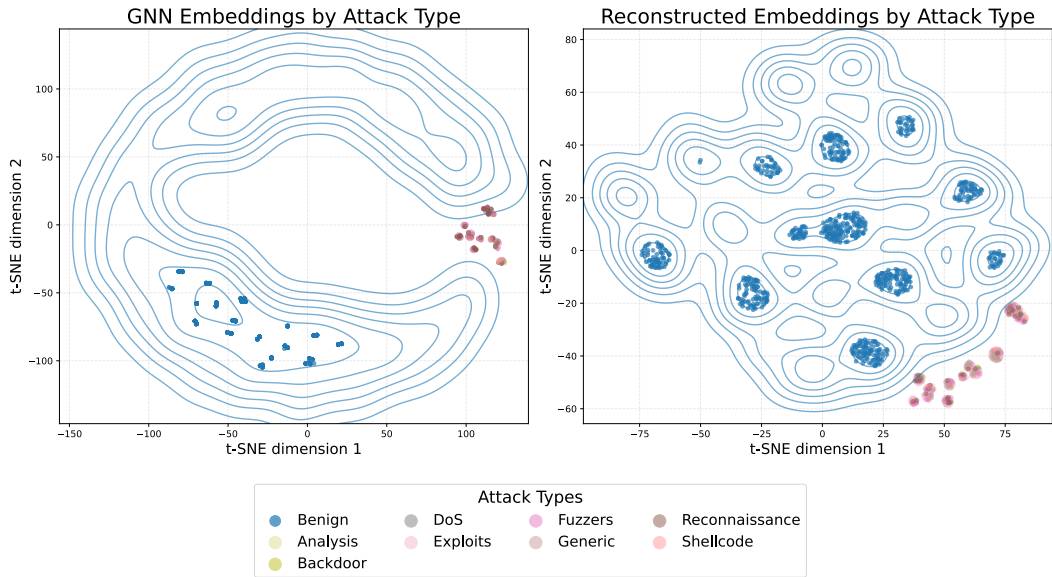


Figure 9: t-SNE visualization of embeddings by attack type in NF-UNSW-NB15-v3, with density contours illustrating the concentration of benign samples.

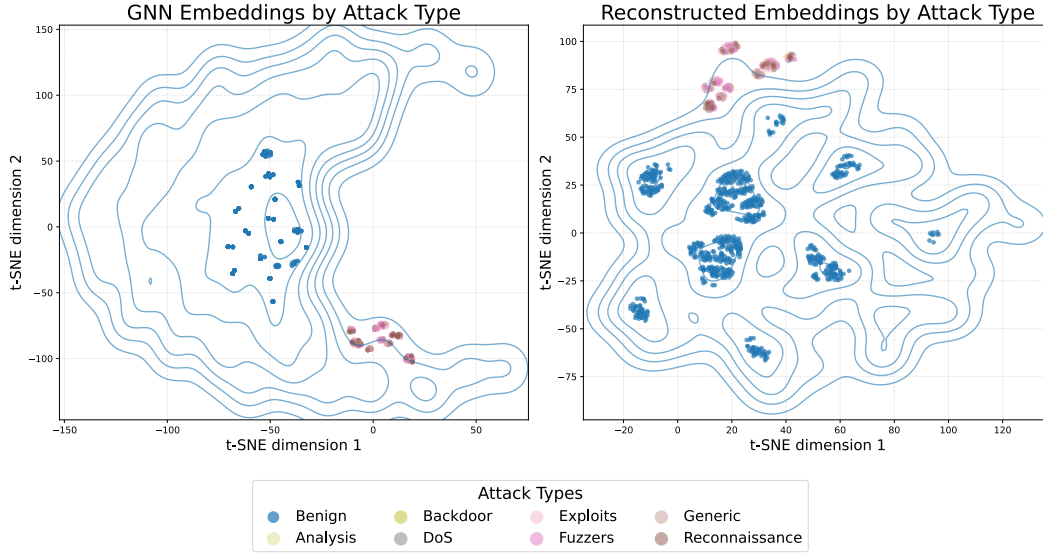


Figure 10: t-SNE visualization of embeddings by attack type in NF-UNSW-NB15-v2.

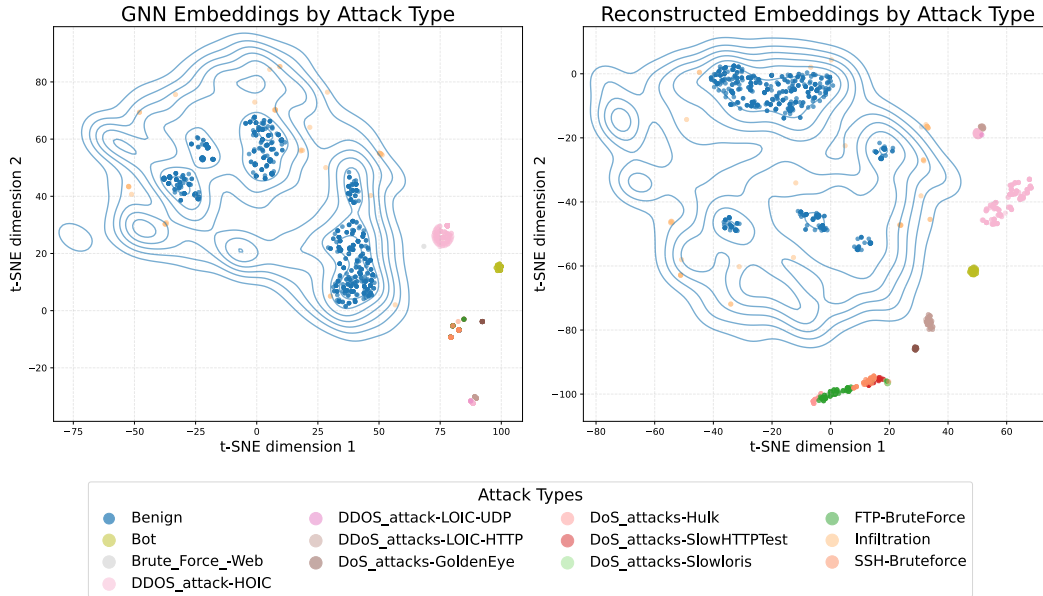


Figure 11: t-SNE visualization of embeddings by attack type in NF-CSE-CIC-IDS2018-v3.

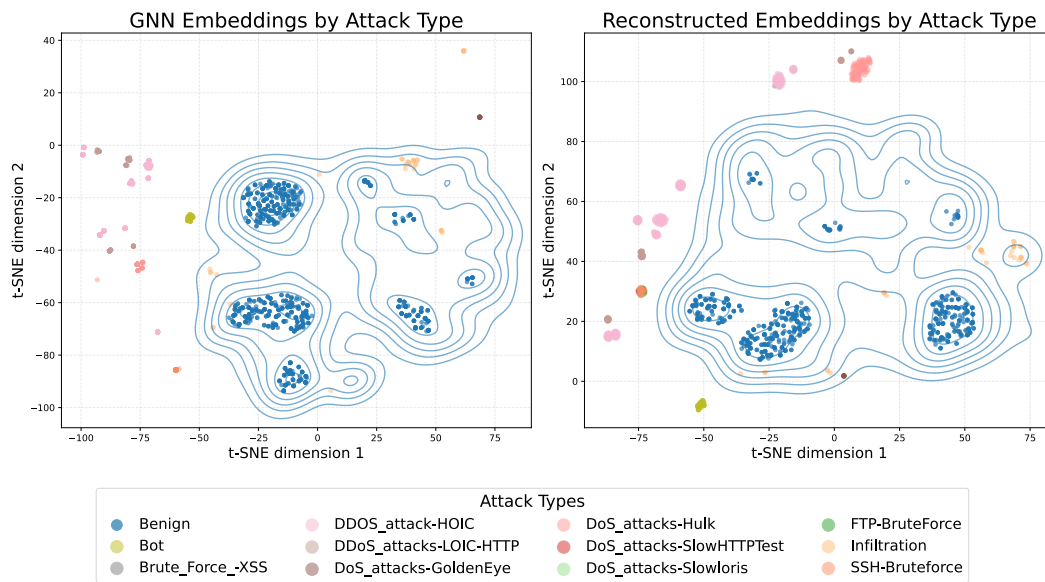


Figure 12: t-SNE visualization of embeddings by attack type in NF-CSE-CIC-IDS2018-v2.

## B.2 Extended Baseline Comparison: Anomal-E and Traditional Methods

In Table 4, we compare the training time and peak GPU memory usage of the evaluated models across datasets. Traditional methods such as CBLOF are excluded from this comparison, as they do not leverage GPU resources.

The results highlight the trade-offs between memory usage, training time, and performance across models. SAFE requires very little GPU memory thanks to its shallow architecture, low input dimensionality, and feature selection, but this comes at the cost of longer training times and substantially lower performance, as shown in Figure 4 and in Table 3. In contrast, Anomal-E’s original design, which relies on batch gradient descent and aggregates over the full neighborhood, leads to a considerable memory footprint of up to 30 GB, making it impractical for large graphs. GraphIDS offers a balanced compromise between efficiency and accuracy, using about 1.37 GB of GPU memory and training in 0.87 hours on average. SimpleAE further reduces compute, training in about 0.76 hours with a peak of roughly 428 MB of GPU memory, while remaining competitive in accuracy in the main tables. Finally, GraphIDS’s mini-batch strategy allows it to learn normal network behavior from the full graphs of the NF-CSE-CIC-IDS2018 datasets without downsampling, which in our experiments led to improved performance and stability.

Tables 5 and 6 summarize the performance of Anomal-E against the set of anomaly detection algorithms introduced in the original paper. These results show that none of these methods, unlike GraphIDS, is able to maintain a consistent performance across all datasets.

In addition, Tables 7 and 8 report results for traditional anomaly detection algorithms applied directly to raw NetFlow features. All these models show substantially lower performance compared to GraphIDS. Among them, CBLOF achieves the most competitive results on average and is used in the main paper as a representative baseline for traditional methods.

Table 4: Comparison of training time and peak GPU memory usage across models.

| Model    | Training Time (h) | Peak Memory (MB) |
|----------|-------------------|------------------|
| SAFE     | $3.59 \pm 1.34$   | 66               |
| Anomal-E | $2.51 \pm 1.39$   | 29,775           |
| T-MAE    | $2.21 \pm 2.87$   | 9,063            |
| SimpleAE | $0.76 \pm 0.65$   | 428              |
| GraphIDS | $0.87 \pm 0.40$   | 1,366            |

Table 5: Performance comparison of different anomaly detection algorithms applied to **Anomal-E** embeddings on the v3 datasets. Results for GraphIDS are included as reference. Bold values indicate statistically significant improvements.

| Model           | Metric   | NF-UNSW-NB15-v3                       | NF-CSE-CIC-IDS2018-v3                 |
|-----------------|----------|---------------------------------------|---------------------------------------|
| Anomal-E-CBLOF  | PR-AUC   | $0.7827 \pm 0.0840$                   | $0.2555 \pm 0.0383$                   |
|                 | Macro F1 | $0.8891 \pm 0.1127$                   | $0.6709 \pm 0.0394$                   |
| Anomal-E-HBOS   | PR-AUC   | $0.8735 \pm 0.0126$                   | $0.1663 \pm 0.0241$                   |
|                 | Macro F1 | $0.9458 \pm 0.0007$                   | $0.5329 \pm 0.0487$                   |
| Anomal-E-IF     | PR-AUC   | $0.7613 \pm 0.0530$                   | $0.1812 \pm 0.0218$                   |
|                 | Macro F1 | $0.9166 \pm 0.0345$                   | $0.5514 \pm 0.0195$                   |
| Anomal-E-PCA    | PR-AUC   | $0.9032 \pm 0.0041$                   | $0.1098 \pm 0.0165$                   |
|                 | Macro F1 | $0.9459 \pm 0.0009$                   | $0.4898 \pm 0.0347$                   |
| GraphIDS (Ours) | PR-AUC   | <b><math>0.9998 \pm 0.0007</math></b> | <b><math>0.8819 \pm 0.0347</math></b> |
|                 | Macro F1 | <b><math>0.9961 \pm 0.0084</math></b> | <b><math>0.9447 \pm 0.0213</math></b> |

Table 6: Performance comparison of different anomaly detection algorithms applied to **Anomal-E** embeddings on the v2 datasets.

| Model           | Metric   | NF-UNSW-NB15-v2                       | NF-CSE-CIC-IDS2018-v2 |
|-----------------|----------|---------------------------------------|-----------------------|
| Anomal-E-CBLOF  | PR-AUC   | $0.7175 \pm 0.0041$                   | $0.9287 \pm 0.0265$   |
|                 | Macro F1 | $0.9262 \pm 0.0008$                   | $0.9410 \pm 0.0161$   |
| Anomal-E-HBOS   | PR-AUC   | $0.7489 \pm 0.0074$                   | $0.9154 \pm 0.0181$   |
|                 | Macro F1 | $0.9156 \pm 0.0217$                   | $0.9415 \pm 0.0131$   |
| Anomal-E-IF     | PR-AUC   | $0.7438 \pm 0.0162$                   | $0.8847 \pm 0.0789$   |
|                 | Macro F1 | $0.9153 \pm 0.0216$                   | $0.9332 \pm 0.0302$   |
| Anomal-E-PCA    | PR-AUC   | $0.7133 \pm 0.0034$                   | $0.9178 \pm 0.0078$   |
|                 | Macro F1 | $0.9262 \pm 0.0008$                   | $0.9436 \pm 0.0076$   |
| GraphIDS (Ours) | PR-AUC   | <b><math>0.8116 \pm 0.0367</math></b> | $0.9201 \pm 0.0238$   |
|                 | Macro F1 | $0.9264 \pm 0.0217$                   | $0.9431 \pm 0.0131$   |

Table 7: Evaluation of **traditional** anomaly detection algorithms on the v3 datasets.

| Model           | Metric   | NF-UNSW-NB15-v3                       | NF-CSE-CIC-IDS2018-v3                 |
|-----------------|----------|---------------------------------------|---------------------------------------|
| CBLOF           | PR-AUC   | $0.3658 \pm 0.0634$                   | $0.2638 \pm 0.0263$                   |
|                 | Macro F1 | $0.7319 \pm 0.0225$                   | $0.6599 \pm 0.0130$                   |
| HBOS            | PR-AUC   | $0.2604 \pm 0.0021$                   | $0.1822 \pm 0.0011$                   |
|                 | Macro F1 | $0.7171 \pm 0.0007$                   | $0.5365 \pm 0.0070$                   |
| IF              | PR-AUC   | $0.2537 \pm 0.0205$                   | $0.1630 \pm 0.0139$                   |
|                 | Macro F1 | $0.6822 \pm 0.0152$                   | $0.5330 \pm 0.0160$                   |
| PCA             | PR-AUC   | $0.4380 \pm 0.0038$                   | $0.1200 \pm 0.0003$                   |
|                 | Macro F1 | $0.7554 \pm 0.0018$                   | $0.5306 \pm 0.0004$                   |
| GraphIDS (Ours) | PR-AUC   | <b><math>0.9998 \pm 0.0007</math></b> | <b><math>0.8819 \pm 0.0347</math></b> |
|                 | Macro F1 | <b><math>0.9961 \pm 0.0084</math></b> | <b><math>0.9447 \pm 0.0213</math></b> |

Table 8: Evaluation of **traditional** anomaly detection algorithms on the v2 datasets.

| Model           | Metric   | NF-UNSW-NB15-v2                       | NF-CSE-CIC-IDS2018-v2                 |
|-----------------|----------|---------------------------------------|---------------------------------------|
| CBLOF           | PR-AUC   | $0.2102 \pm 0.0157$                   | $0.7822 \pm 0.0198$                   |
|                 | Macro F1 | $0.7046 \pm 0.0140$                   | $0.8889 \pm 0.0068$                   |
| HBOS            | PR-AUC   | $0.3197 \pm 0.0036$                   | $0.6662 \pm 0.0205$                   |
|                 | Macro F1 | $0.7032 \pm 0.0012$                   | $0.8578 \pm 0.0034$                   |
| IF              | PR-AUC   | $0.1914 \pm 0.0075$                   | $0.6124 \pm 0.0269$                   |
|                 | Macro F1 | $0.6844 \pm 0.0071$                   | $0.8321 \pm 0.0099$                   |
| PCA             | PR-AUC   | $0.2840 \pm 0.0039$                   | $0.5911 \pm 0.0014$                   |
|                 | Macro F1 | $0.6975 \pm 0.0023$                   | $0.6128 \pm 0.0076$                   |
| GraphIDS (Ours) | PR-AUC   | <b><math>0.8116 \pm 0.0367</math></b> | <b><math>0.9201 \pm 0.0238</math></b> |
|                 | Macro F1 | <b><math>0.9264 \pm 0.0217</math></b> | <b><math>0.9431 \pm 0.0131</math></b> |



## C Ablation Studies

In this section, we aim to isolate the individual contributions of specific design choices in GraphIDS. To reduce the computational cost of the ablation study, we conducted experiments over fewer random seeds than those used for the main results. Nevertheless, the setup was sufficient to clearly identify the impact of each component.

### C.1 Effect of Timestamp Features

In this ablation study, we evaluate the impact of including timestamps (FLOW\_START\_MILLISECONDS and FLOW\_END\_MILLISECONDS) among the input features. Our goal is to determine whether they provide useful temporal information or only introduce noise. As shown in Table 9, their inclusion has negligible impact on NF-UNSW-NB15-v3, but clearly degrades the performance on NF-CSE-CIC-IDS2018-v3. This suggests that, overall, excluding timestamps leads to more efficient representations.

Table 9: Effect of including timestamp features on model performance across v3 datasets.

| Model  | Metric   | NF-UNSW-NB15-v3     | NF-CSE-CIC-IDS2018-v3 |
|--------|----------|---------------------|-----------------------|
| w/ TS  | PR-AUC   | $0.9991 \pm 0.0011$ | $0.7909 \pm 0.0151$   |
|        | Macro F1 | $0.9957 \pm 0.0091$ | $0.9088 \pm 0.0110$   |
| w/o TS | PR-AUC   | $0.9989 \pm 0.0017$ | $0.8523 \pm 0.0283$   |
|        | Macro F1 | $0.9982 \pm 0.0029$ | $0.9385 \pm 0.0122$   |

### C.2 Effect of Positional Encoding

To investigate whether our model can benefit from sequence modeling, beyond co-occurrence patterns, we temporally ordered the flows in the v3 datasets and added positional encodings to each input window before passing it to the Transformer. We evaluated two variants: sinusoidal and learnable encodings.

For sinusoidal positional encoding, we used the formulation from [27], where the encoding at position  $pos$  and dimension  $i$  is given by:

$$PE_{(pos, 2i)} = \sin\left(\frac{pos}{10000^{2i/d}}\right), \quad PE_{(pos, 2i+1)} = \cos\left(\frac{pos}{10000^{2i/d}}\right) \quad (3)$$

For learnable positional encoding, we used a parameter matrix  $\mathbf{P} \in \mathbb{R}^{L \times d}$ , with  $L$  as the maximum sequence length and  $d$  the embedding dimension. This matrix is optimized along with the rest of the model during training.

The results in Table 10 indicate that positional encodings have little impact on GraphIDS’s performance, suggesting that the model primarily learns global co-occurrence patterns rather than temporal dependencies. To reflect this, we omit positional encodings in the main experiments and shuffle the flow order.

Table 10: Effect of positional encoding across datasets.

| Model                   | Metric   | NF-UNSW-NB15-v3     | NF-CSE-CIC-IDS2018-v3 |
|-------------------------|----------|---------------------|-----------------------|
| w/ Learnable Encoding   | PR-AUC   | $0.9939 \pm 0.0126$ | $0.8572 \pm 0.0284$   |
|                         | Macro F1 | $0.9873 \pm 0.0263$ | $0.9480 \pm 0.0153$   |
| w/ Sinusoidal Encoding  | PR-AUC   | $0.9955 \pm 0.0110$ | $0.8537 \pm 0.0421$   |
|                         | Macro F1 | $0.9904 \pm 0.0214$ | $0.9446 \pm 0.0171$   |
| w/o Positional Encoding | PR-AUC   | $0.9955 \pm 0.0126$ | $0.8661 \pm 0.0411$   |
|                         | Macro F1 | $0.9821 \pm 0.0256$ | $0.9546 \pm 0.0099$   |

### C.3 Effect of GNN Dropout Rate

As shown in Table 11, the dropout rate in E-GraphSAGE has a relevant impact on GraphIDS’s overall performance. Higher dropout rates achieved better results on the NF-UNSW-NB15 datasets, suggesting that the GNN is prone to overfitting in smaller network environments. In these cases, stronger regularization helps the model generalize to unseen data. On the NF-CSE-CIC-IDS2018 datasets, introducing a non-zero dropout rate also helped stabilize the learning process, although its impact on final performance was less pronounced.

Table 11: Effect of the GNN dropout rate across datasets.

| Dataset               | Metric   | 0.0    | 0.25   | 0.5    | 0.6    | 0.7    |
|-----------------------|----------|--------|--------|--------|--------|--------|
| NF-UNSW-NB15-v3       | PR-AUC   | 0.9773 | 0.9619 | 0.9845 | 0.9998 | 0.9790 |
|                       | Macro F1 | 1.0000 | 0.9999 | 1.0000 | 1.0000 | 1.0000 |
| NF-CSE-CIC-IDS2018-v3 | PR-AUC   | 0.8758 | 0.8645 | 0.8630 | 0.8461 | 0.8621 |
|                       | Macro F1 | 0.9466 | 0.9270 | 0.9450 | 0.9160 | 0.9219 |
| NF-UNSW-NB15-v2       | PR-AUC   | 0.8117 | 0.8154 | 0.8129 | 0.8094 | 0.8301 |
|                       | Macro F1 | 0.9342 | 0.9303 | 0.9140 | 0.9241 | 0.9285 |
| NF-CSE-CIC-IDS2018-v2 | PR-AUC   | 0.8952 | 0.8964 | 0.9005 | 0.8886 | 0.8925 |
|                       | Macro F1 | 0.9322 | 0.9429 | 0.9411 | 0.9402 | 0.9396 |

### C.4 Effect of Masking Ratio

We found that the masking ratio had a noticeable impact on the model’s performance for NF-CSE-CIC-IDS2018-v3. In this case, a masking ratio of 0.15 made the reconstruction task sufficiently challenging for GraphIDS to learn more complex relationships within the flow embeddings. We also noticed that ratios of 0.7 or higher led to gradient explosions and training instability across all datasets.

Table 12: Effect of the attention mask ratio across datasets.

| Dataset               | Metric   | 0.0    | 0.15   | 0.3    | 0.5    | 0.7    |
|-----------------------|----------|--------|--------|--------|--------|--------|
| NF-UNSW-NB15-v3       | PR-AUC   | 0.9992 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
|                       | Macro F1 | 0.9998 | 0.9999 | 0.9793 | 0.9998 | 0.9998 |
| NF-CSE-CIC-IDS2018-v3 | PR-AUC   | 0.8659 | 0.8772 | 0.8691 | 0.8391 | 0.8216 |
|                       | Macro F1 | 0.9445 | 0.9472 | 0.9485 | 0.9051 | 0.9308 |
| NF-UNSW-NB15-v2       | PR-AUC   | 0.8373 | 0.8384 | 0.8384 | 0.8261 | 0.8289 |
|                       | Macro F1 | 0.9344 | 0.9342 | 0.9344 | 0.9299 | 0.8756 |
| NF-CSE-CIC-IDS2018-v2 | PR-AUC   | 0.9115 | 0.9133 | 0.9132 | 0.9132 | 0.9155 |
|                       | Macro F1 | 0.9390 | 0.9419 | 0.9419 | 0.9398 | 0.9414 |

### C.5 Effect of Neighborhood Size

We explored various neighborhood sizes during the initial development phase, as this choice directly impacts both model performance and computational cost. To rigorously validate our design, we conducted an ablation study comparing 1-hop, 2-hop, and 3-hop variants of GraphIDS under identical experimental conditions. As shown in Table 13, increasing the number of hops did not consistently improve performance, while substantially increasing training and inference runtime by up to  $3\times$  and, on average, using 24% more memory. This not only makes extensive hyperparameter tuning impractical but also negatively impacts real-time latency, a critical aspect of intrusion detection.

The results show that the 1-hop configuration delivers strong and stable performance across all datasets. While the 3-hop variant scores slightly higher on NF-CSE-CIC-IDS2018-v3, the difference is not statistically significant. On other datasets, larger neighborhoods lead to degraded or less stable performance, suggesting that increasing the receptive field may introduce noise from distant,

less relevant nodes, thereby diluting local information. Given the added computational overhead, these results support our choice of a 1-hop neighborhood as an effective and efficient default. Our implementation, however, supports arbitrary  $n$ -hop configurations.

Table 13: Effect of neighborhood size (number of GNN hops) across datasets. Mean (std) over seeds.

| Dataset               | Metric   | 1-hop               | 2-hop               | 3-hop               |
|-----------------------|----------|---------------------|---------------------|---------------------|
| NF-UNSW-NB15-v3       | PR-AUC   | $0.9998 \pm 0.0007$ | $0.9980 \pm 0.0018$ | $0.9992 \pm 0.0008$ |
|                       | Macro F1 | $0.9961 \pm 0.0084$ | $0.9935 \pm 0.0142$ | $0.9999 \pm 0.0001$ |
| NF-CSE-CIC-IDS2018-v3 | PR-AUC   | $0.8819 \pm 0.0347$ | $0.8385 \pm 0.0528$ | $0.8969 \pm 0.0325$ |
|                       | Macro F1 | $0.9447 \pm 0.0213$ | $0.9370 \pm 0.0194$ | $0.9606 \pm 0.0083$ |
| NF-UNSW-NB15-v2       | PR-AUC   | $0.8116 \pm 0.0367$ | $0.7883 \pm 0.0321$ | $0.8238 \pm 0.0521$ |
|                       | Macro F1 | $0.9264 \pm 0.0217$ | $0.9147 \pm 0.0076$ | $0.8005 \pm 0.2486$ |
| NF-CSE-CIC-IDS2018-v2 | PR-AUC   | $0.9201 \pm 0.0238$ | $0.7539 \pm 0.3555$ | $0.7482 \pm 0.3526$ |
|                       | Macro F1 | $0.9431 \pm 0.0131$ | $0.8371 \pm 0.2071$ | $0.8495 \pm 0.2133$ |

## D Hyperparameters

Table 14 reports the complete set of optimized hyperparameters used for the GraphIDS model. While these were selected to maximize PR-AUC performance, memory usage can be reduced by decreasing the Transformer’s window size, while the GNN remains efficient by randomly sampling subsets of neighboring edges.

Table 14: Hyperparameters for the GraphIDS model across datasets. Dataset names are shortened for formatting.

| Parameter                     | UNSW-NB15-v3       | NF-CSE-CIC-IDS2018-v3 | UNSW-NB15-v2         | NF-CSE-CIC-IDS2018-v2 |
|-------------------------------|--------------------|-----------------------|----------------------|-----------------------|
| <i>GNN Parameters</i>         |                    |                       |                      |                       |
| edim_out                      | 96                 | 64                    | 72                   | 64                    |
| nhops                         | 1                  | 1                     | 1                    | 1                     |
| fanout                        | 32,768             | 32,768                | 32,768               | 32,768                |
| agg_type                      | mean               | mean                  | mean                 | mean                  |
| dropout                       | 0.6                | 0.5                   | 0.75                 | 0.5                   |
| <i>Transformer Parameters</i> |                    |                       |                      |                       |
| num_layers                    | 1                  | 1                     | 1                    | 1                     |
| embed_dim                     | 48                 | 32                    | 48                   | 32                    |
| window_size                   | 512                | 512                   | 512                  | 512                   |
| mask_ratio                    | 0.15               | 0.15                  | 0.15                 | 0.15                  |
| dropout                       | 0.0                | 0.2                   | 0.0                  | 0.2                   |
| <i>Training Parameters</i>    |                    |                       |                      |                       |
| learning_rate                 | $1 \times 10^{-4}$ | $1 \times 10^{-4}$    | $1.1 \times 10^{-5}$ | $7.4 \times 10^{-5}$  |
| gnn_weight_decay              | 0.6                | 0.6                   | 0.6                  | 0.6                   |
| ae_weight_decay               | 0.04               | 0.04                  | 0.046                | 0.011                 |
| gnn_batch_size                | 16,384             | 16,384                | 32,768               | 16,384                |
| ae_batch_size                 | 64                 | 64                    | 64                   | 64                    |