**The Intellecta International Journal of Research and Innovation**
Volume 1 Issue 1 (Oct-Dec 2026) | e-ISSN: Applied | Page No: 45-61.

# PROTOCOL-STRATIFIED ANOMALY DETECTION OF TCP, UDP AND ICMP ATTACKS WITH A LINEAR-ATTENTION TRANSFORMER: EVIDENCE FROM NSL-KDD AND UNSW-NB15

**Dr. Archana**
*Department of Computer Science*
*Mumbai, MH India*
*Email id: archcse123@gmail.com*

## ABSTRACT

*In general, studies of intrusion detection report only one accuracy or F1 score calculated on the entire test set. Since TCP, UDP and ICMP toxicate attack traffic in very different ratios, an average such a figure overprofiles problems of varying difficulty and may mask a detector that collapses on a complete protocol. We introduce a protocol-aware anomaly detector, based on the Linformer, a variation of transformer that scales its attention cost linearly with the input length, and we rank it on each protocol on two benchmark corpora. On NSL-KDD (148,517 records) and UNSW-NB15 (215,757 records after protocol filtering), six experiments are done on a fully crossed design comparing the Linformer with a Decision Tree and a random forest that has 72,856 held-out records. On NSL-KDD the Linformer reached an accuracy of 0.9823, an F1-score of 0.9816 and an AUC-ROC of 0.9987; on UNSW-NB15 the corresponding values were 0.9231, 0.9344 and 0.9832. Breaking down the results according to protocol revealed a bug that the cumulative figures disguised: only 3.61 percent of the ICMP attacks were identified by the Random Forest, which has a cumulative accuracy of 0.8784 on NSL-KDD, compared to 99.30 percent by the Linformer. The protocol subsets (among five that are assessable) F1 increased with attack prevalence (Pearson = 0.7773), indicating that a direct solution is per-protocol threshold calibration. Extrapolating the measured error rates onto the prevalence values determined that accuracy reduces to 0.3415 at 1% attack traffic, indicating the detector should not be used as a blocker, despite the fact that at 0.9793, the prevalence matches the benchmark prevalence criterion.*

***Keywords:*** *Anomaly detection · Intrusion detection system · Linformer · Protocol-stratified evaluation · NSL-KDD · UNSW-NB15*

## 1. INTRODUCTION

The use of network intrusion detection has shifted towards non-signature (anomaly) detection. A signature engine uses only the information that has already been catalogued, and an anomaly detector learns the normal appearance of traffic and indicates departures, maintaining the previously unknown behaviour of attack yet still accessible (Denning 1987; Bace and Mell 2001; Liao et al. 2013). Majority of these detectors are currently constructed by using machine learning. A list of tree ensembles, support vector machines, autoencoders and recurrent networks with benchmark traffic data training (Buczak and Guven 2016; Pang et al. 2021), and new reviews indicate that the field is shifting toward deep and attention-based models (Chinnasamy et al. 2025; Pinto Neto et al. 2025).

The results of this literature inform a practitioner only to a certain extent because of two habits of the literature. The former is the aggregate reporting. Most studies give one accuracy or F1 value computed over an entire evaluation set (Buczak and Guven 2016). Transport and control protocols the network traffic is carried on offer varying threat surfaces: TCP is connection oriented and the vehicle of SYN flooding and port scanning, UDP is connectionless and

the vehicle of amplification and flood, and ICMP is a diagnostic protocol that has minimal legitimate traffic, but is used extensively in sweep and flood. In NSL-KDD the proportion of attack records is 49.5% in TCP, 19.3% in UDP and 85.0 atICMP. Aggregate measure is an average of three problems with varying base rates. The most prevalent protocol dominates that average, so a detector is able to record a respectable score and fail to succeed on a minority protocol. According to Ring et al. (2019), the composition of a protocol varies across corpora, but this variability is often not reported on evaluation.

The second practice is to halt at benchmark accuracy. The prevalences of attacks is close to half in benchmark corpora, compared to a much lower fraction of attacks in flows on an operational network. Precision is reliant on prevalence, such a figure measured on a benchmark overstates accuracy that a deployed detector will provide. That was a base-rate problem as described by Axelsson (2000) over 20 years ago, with Sommer and Paxson (2010) asserting machine-learning results in intrusion detection seldom withstands examination of the aspect in which they have been tested.

We accept the two points. It is based on fast persistent contrastive divergence deep belief network (FPCD-DBN) of Jeong et al. (2024), with a reported accuracy of 0.8939 on NSL-KDD and 0.8662 on UNSW-NB15 and substitutes the deep belief network with a Linformer (Wang et al. 2020). Self-attention is a pairwise interaction between the inputs explicitly (Vaswani et al. 2017), and numerous attacks are manifested in the form of an odd number of combinations of several traffic properties, not a single extreme value. This pairwise perspective of the Linformer is retained, with the quadratic cost of ordinary attention being eliminated at network volumes. The contributions are as follows.

- A protocol-sensitive Linformer detector of TCP, UDP as well as ICMP traffic, compared to two tree-based baselines, introduced under equal features, partitions, metrics and decision threshold.
- Techniques: A protocol-stratified assessment which recalculates all the measures in each protocol block, and leaves undefined measures undefined, not zero, and reveals the fact that overall measures masked an almost complete ICMP failure of the Random Forest baseline.
- Two corpora provide evidence that per-protocol threshold calibration is a concrete improvement by! detecting the quality of the attacks by each combination of protocols used.
- Alternative: A simulation of the observed sensitivity and specificity onto realistic prevalence of operational attacks, as well as local interpretations of individual alerts, to demonstrate what the detector would imply on realistic analyst team work.

Section 2 presents related work. Section 3 explains the data and preprocessing, models and evaluation protocol. Section 4 presents the results, section 5 presents results in discussion along with the study limitations and section 6 concludes.

## 2. RELATED WORK

Prone to early intrusion detection research was dominating by KDD Cup 1999 data, which introduced redundancy in records that skewed the analysis to frequent connections that occurred repeatedly. Tavallaee et al. (2009) eliminated the redundancy to form NSL-KDD, and put in the test division of this algorithm the attack signatures that are neither found in the training division nor in the test division. UNSW-NB15 was modeled using IXIA PerfectStorm to ensure the assurance of capturing both benign and attack behaviour in contemporary times and then compared with the statistics of the KDD family (Moustafa and Slay 2016). Ring et al. (2019), surveyed network intrusion datasets and found out that they have certain limitations but so far the only foundation of comparative claims is the available public benchmarks.

Both corpora have been subjected to deep-models. Tang et al. (2016) suggested an accuracy of 0.7575 with a deep neural network, Yin et al. (2017) suggested 0.8129 with a recurrent network, Aggarwal and Sharma (2015) suggested 0.8621 with random-tree analysis of class-wise attributes, Aygun and Yavuz (2017) suggested 0.8828 and 0.8865 with an autoencoder and a denoising autoencoder, and Naseer et al. (2018) suggested 0.8900 with a long short-term memory network on NSL-KDD. On UNSW-NB15, the authors reported 0.7639 with a naive Bayes, Malaiya et al. (2019) reported 0.8190 with a support vector machine, Zhou et al. (2020) reported 0.8363 with a random forest and

Min et al. (2021) reported 0.8530 with a memory-augmented autoencoder. Jeong et al. (2024) obtained the following results unified via a common protocol and proposed the FPCD-DBN, incorporating a fast-weight mechanism of Tieleman and Hinton (2009) with persistent contrastive divergence (Berglund and Raiko 2013) and achieved 0.8939 on NSL-KDD and 0.8662 on UNSW-NB15.

Anomaly detection has become a focus, as well. The Vaswani et al. (2017) transformer is score based and Xu et al. (2022) demonstrated that the structure of association that was learned by attention detects abnormal behaviour compared to normal behaviour in sequential data. The cost of attention increases with the square of the input length; indeed, the Linformer of Wang et al. (2020) formats keys and values into a constant lower-dimensional space and removes the cost to a linear cost. Fawcett (2006), Powers (2011), and Chicco and Jurman (2020) respectively have presented how ROC is interpreted, how many measures should be reported simultaneously, and how Matthews correlation coefficient (MCC) is more informative than the accuracy or the F1, when the classes can be unbalanced. To elaborate, a local linear surrogate of LIME by Ribeiro et al. (2016) is the one trained on a single prediction.

Pressures on accuracy on these corpora have been increased since 2024. Xi et al. (2024) developed IDS-MTran which is a multi-scale transformer with multi-kernel convolutions and found it to perform at over 99% accuracy in NSL-KDD, CIC-DDoS2019 and UNSW-NB15. Umer et al. (2025) selected wrapper-based features with multi-head attention transformer against UNSW-NB15, and Farhan et al. (2025) achieved binary accuracy of 97.93% with a deep network which inputs eight features selected by an Extra Trees selector on the same corpus. A comparison of the Decision Tree, Regular Forest, SVM and XGBoost with UNSW-NB15 through five-fold stratified cross-validation revealed that XGBoost performed the best (accuracy 0.97, AUC 0.98) with sttl, sload and dload topping the list of influential features. Such studies split according to their own discretion, and set of features and schemes of validation, so their numbers cannot be directly compared one to another or with ours. They all de-spool transport protocol.

In nearly all of this literature, there is little to no protocol break down in the results. All studies included in the table compiled by Jeong et al. (2024) do not specify their per-protocol result, none of them indicate the extent to which their precision would increase at operational prevalence, and not very many provide the description of individual alerts. The given study attempts to fill in these three gaps and make its comparison of the results with the previous ones open to test.

## 3. MATERIALS AND METHODS

### *3.1 Research design*

The study follows a positivist, deductive and experimental design. It is a fully crossed factorial experiment with two factors: the detection model (Decision Tree, Random Forest, Linformer) and the benchmark corpus (NSL-KDD, UNSW-NB15). The six resulting experiments, E1 to E6, are all populated, so no comparison needs interpolation. Transport protocol was not manipulated. It defines the subgroups within which every metric is recomputed. The feature matrix, the partitions, the metric implementation, the decision threshold of 0.5 and the random seed of 42 were held constant across all six experiments, so a difference in outcome points to the model or the corpus and not to the setup.

### *3.2 Datasets*

Both corpora were obtained from public repositories (NSL-KDD files KDDTrain+ and KDDTest+; UNSW-NB15 training and testing CSV files) and stored in Google Drive for use in a Colaboratory runtime. Table 1 compares them. NSL-KDD has all three protocols in useful volume, which makes it the main corpus for the per-protocol analysis. UNSW-NB15 supplies contemporary attack behaviour and a richer flow-level vocabulary but contains only 15 ICMP records, so every ICMP claim in this paper rests on NSL-KDD. In the distribution used, the UNSW-NB15 file labelled as the training set holds 82,332 records and the file labelled as the testing set holds 175,341; because the two files were merged before partitioning, this naming has no effect on the results.

**Table 1 Characteristics of the two benchmark corpora**

| Characteristic | NSL-KDD | UNSW-NB15 |
|---|---|---|
| Total records | 148,517 | 257,673 |
| Schema columns | 43 | 44 |
| Feature orientation | Connection level | Flow level |
| Distinct protocol values | 3 | 133 |
| Attack granularity | 39 named signatures | 9 categories |
| Normal / attack proportion (%) | 51.88 / 48.12 | 36.09 / 63.91 |
| TCP / UDP / ICMP records | 121,569 / 17,614 / 9,334 | 123,041 / 92,701 / 15 |
| Missing values | None | None |
| Features retained after preprocessing | 27 | 42 |

The attack taxonomy of NSL-KDD is severely long tailed. The TCP SYN flood signature *neptune* accounts for 45,871 records, or 64.19% of the attack class, while *spy*, *worm*, *sqlattack* and *udpstorm* contribute two records each. Four signatures make up more than 80% of attack traffic. Tuning a classifier for aggregate accuracy thus amounts to tuning it for SYN flooding. Attack prevalence also differs sharply by protocol (Fig. 1): TCP is close to balanced, UDP is attack scarce and ICMP is attack dominant, which means a majority-class rule would already reach 80.7% accuracy on UDP and 85.0% on ICMP. Inside these subsets accuracy says little; precision, recall and MCC carry the interpretation.

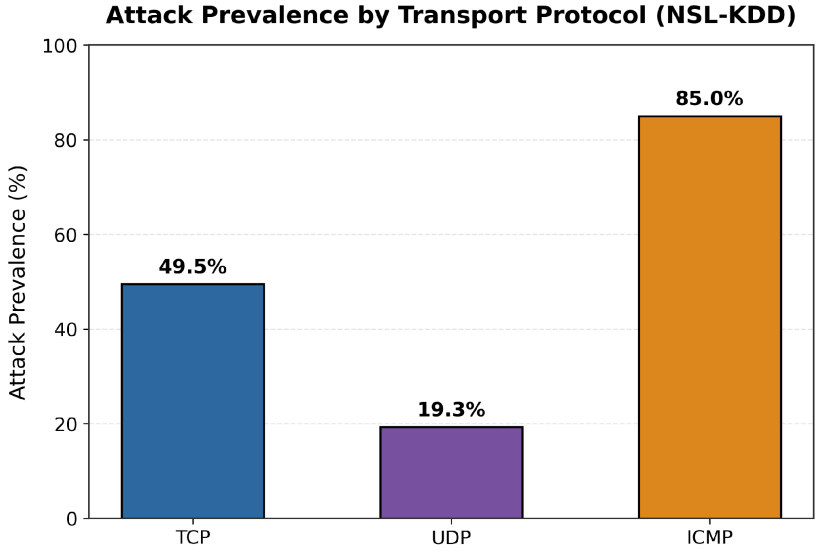



*Fig. 1 Attack prevalence within each transport protocol in NSL-KDD. The three protocols pose three different learning problems: balanced (TCP), attack scarce (UDP) and attack dominant (ICMP)*

### *3.3 Preprocessing*

The two corpora were read with a schema explicitly given and were audited against missing values, duplicate and constant columns. There were no missing values. NSL-KDD has 610 pairs of verbatim copies (0.41 of the corpus), which were kept to ensure that the corpus being analysed matches that given out, and a single zero-variance feature, the number of outbound commands, which was dropped. The only records available were TCP, UDP and ICMP. This filter made NSL-KDD appear the same way and trimmed UNSW-NB15 down to 215,757 records.

The multiclass data were raised to two-class target (0 = normal, 1 = attack). The measurement of a multiclass formulation would consist of two-example signatures, the output of which cannot be generalized to come to a decision, and the question of operation is whether the traffic should be researched, not what its historical name is. UNSW-NB15 includes an extra binary label column of attack_cat that was removed during the creation of the feature matrix: this would have given each model a copy of the target.

Nominal attributes were lexicographically coded as labels. Since the protocol field is an input and the grouping variable to be stratified, its encoding was checked: icmp corresponds to 0, tcp to 1 and udp to 2 and delayed subsets sizes corresponding to the protocol shares in NSL-KDD (1,872, 24,314 and 3,518) are the same as the numbers produced by protocol shares. In the case of NSL-KDD, all 27 attributes were retained: the encoded protocol, basic connection attributes, two-second window traffic attributes and host-based traffic attributes. Attributes of payload contents were not included since the study deals with a protocol level behaviour. In the case of UNSW-NB15, each attribute except the record identifier (and both label columns) were retained resulting in 42. All attributes were normalised to have a mean of zero and unit variance, with in NSL-KDD, the mean of src-bytes is 40227.95 and the standard deviation is 5.41 million, and the serror-rate is in a unit interval. Each corpus was divided 80/20 into training and held-out partitions with fixed random state, and a stratified subset 20% of training partition was saved-off, stratified by class, as validation. The result is summarised in table 2.

**Table 2 Outcome of the preprocessing pipeline**

| Outcome | NSL-KDD | UNSW-NB15 |
|---|---|---|
| Records after protocol filtering | 148,517 | 215,757 |
| Attributes presented to the models | 27 | 42 |
| Training partition | 118,813 | 172,605 |
| Validation records drawn from training | 23,763 | 34,521 |
| Held-out partition | 29,704 | 43,152 |
| Held-out TCP / UDP / ICMP | 24,314 / 3,518 / 1,872 | 24,596 / 18,550 / 6 |

### *3.4 Baseline classifiers*

The Decision Tree follows the CART formulation of Breiman et al. (1984) as implemented in Scikit-learn (Pedregosa et al. 2011), with a maximum depth of 2, a minimum of 100 records to split, a minimum of 50 records per leaf, 30% of features considered at each split and balanced class weights. The fitted tree has four leaves. The Random Forest (Breiman 2001) combines five trees of depth 2 with the same split and leaf limits and no class reweighting. We kept both baselines small on purpose, as fully readable reference points. Section 5 discusses what that choice costs.

### *3.5 Proposed Linformer detector*

The detector is shown in Fig. 2. The standardised attribute vector passes through batch normalisation (Ioffe and Szegedy 2015) and is projected to a model dimension of 256 by a linear layer followed by layer normalisation (Ba et al. 2016), a GELU activation (Hendrycks and Gimpel 2016) and dropout (Srivastava et al. 2014). Three encoder layers follow. Each combines an eight-head linear attention module with a position-wise feed-forward network of width 256, with residual connections and layer normalisation around both. A classification head reduces the representation to two logits, and the softmax probability of the attack class is taken as the anomaly score.

Within the attention module, queries, keys and values are obtained by learned linear maps and split into heads. A kernel projection with projection ratio 4 maps queries and keys into a reduced dimension, and a ReLU feature map φ keeps them non-negative. The output comes from two contractions, and no explicit attention matrix is built:

$$\mathrm{Attn}(Q, K, V) = \varphi(Q)\,[\varphi(K)^{\mathrm{T}}\,V] \,/\, \varphi(Q)\,[\varphi(K)^{\mathrm{T}}\,\mathbf{1}] \qquad (1)$$

The size of the bracketed summary depends on the projection dimension, not on the input length, and the denominator is clamped from below to avoid division by a vanishing sum. Because no matrix with both dimensions scaling with input length is formed, memory and computation grow linearly (Wang et al. 2020). The model has 1,294,008 trainable parameters for NSL-KDD and 1,297,878 for UNSW-NB15; the difference of 3,870 comes entirely from the input projection.

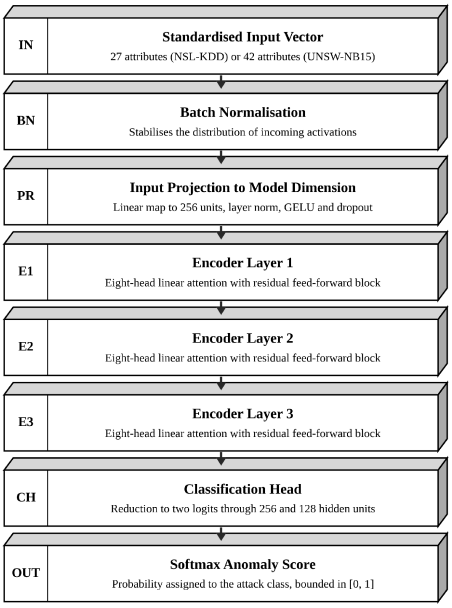



*Fig. 2 Architecture of the proposed Linformer detector, from the standardised input vector to the softmax anomaly score*

The Linformer was preferred to the deep belief network of Jeong et al. (2024) for three reasons. Attacks such as a SYN flood show up as a conjunction of a high connection count, a high error rate and a near-zero reverse byte count, and attention represents such interactions explicitly, whereas a stack of restricted Boltzmann machines (Fischer and Igel 2012; Hinton 2009) captures them only implicitly. Linear attention removes the quadratic cost of standard attention. And end-to-end training against a differentiable objective makes convergence visible in the loss curves, which layerwise contrastive divergence does not.

Training minimised cross-entropy with the Adam optimiser (Kingma and Ba 2015) in PyTorch (Paszke et al. 2019) at a learning rate of 0.001, a batch size of 128 and dropout of 0.1, for a fixed budget of 20 epochs without early stopping. Loss, accuracy and F1 were logged on the training and validation partitions after every epoch. No CUDA device was available, so all models were trained on CPU. Seeds were fixed at 42 for NumPy, PyTorch initialisation, partitioning and both tree baselines; only the order of mini-batches within an epoch was left random.

### *3.6 Evaluation protocol*

Five primary metrics were computed for every model: accuracy, precision, recall, F1 and the area under the ROC curve (AUC). Four derived quantities were added: specificity, false positive rate (FPR), false negative rate (FNR) and MCC, defined as

$$\mathrm{MCC} = (\mathrm{TP}\cdot\mathrm{TN} - \mathrm{FP}\cdot\mathrm{FN}) / \sqrt{[(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})]} \quad (2)$$

where TP, TN, FP and FN are the confusion-matrix counts. Precision governs the alert burden on the analyst, recall governs exposure to undetected compromise, and AUC measures the quality of the ranking induced by the anomaly score across all thresholds (Fawcett 2006). When a model has weak discrete metrics but a high AUC, its threshold is the problem, not its representation. All metrics were computed with a single library implementation. Every metric was computed twice: once on the whole held-out partition and once within each protocol subset, using a Boolean mask built from the verified protocol encoding and applied jointly to the true labels, predictions and scores. Where a subset holds only one class, AUC is undefined, and where it holds no positives, precision and recall are undefined; such cells are reported as undefined, not as zero. Local explanations were produced with LIME (Ribeiro et al. 2016) for one record per protocol, and global feature importances were taken from the tree baselines as an independent check.

**The Intellecta International Journal of Research and Innovation**
Volume 1 Issue 1 (Oct-Dec 2026) | e-ISSN: Applied | Page No: 45-61.

# 4. RESULTS

## *4.1 Baseline classifiers and the concealed ICMP failure*

Table 3 gives the aggregate results of the two baselines. On NSL-KDD neither dominates. The Decision Tree has the higher accuracy (0.9020 against 0.8784), recall (0.8678 against 0.7626) and F1 (0.8947 against 0.8575), while the Random Forest has the higher precision (0.9795) and AUC (0.9649). The Random Forest ranks records better but converts that ranking into worse decisions, which points to a poorly placed threshold. On UNSW-NB15 the Decision Tree reaches a recall of exactly 1.0000 with a precision of 0.7939. A model that flagged every record would have recall 1.0000 and precision 0.6391, the attack prevalence of the corpus; the Decision Tree is close to that pattern, helped by its balanced class weights.

**Table 3 Aggregate performance of the baseline classifiers on the held-out partitions**

| Metric | DT (NSL-KDD) | RF (NSL-KDD) | DT (UNSW-NB15) | RF (UNSW-NB15) |
|---|---|---|---|---|
| Accuracy | 0.9020 | 0.8784 | 0.8481 | 0.8849 |
| Precision | 0.9234 | 0.9795 | 0.7939 | 0.8988 |
| Recall | 0.8678 | 0.7626 | 1.0000 | 0.9051 |
| F1-score | 0.8947 | 0.8575 | 0.8851 | 0.9020 |
| AUC-ROC | 0.9240 | 0.9649 | 0.9225 | 0.9652 |

DT Decision Tree, RF Random Forest

Stratification changes the picture (Table 4, Fig. 3). On TCP both baselines are competent, with F1 values of 0.9191 and 0.9217. On the NSL-KDD ICMP subset, however, the Random Forest reaches an accuracy of 0.1848, a recall of 0.0361 and an F1 of 0.0695: it detects fewer than four of every hundred ICMP attacks. Its accuracy is far below the 0.8424 that the trivial rule of flagging every ICMP record would achieve. The shallow ensemble learned a boundary calibrated to the near-balanced prior of the whole corpus and applied it to a subset whose prior is inverted. The Decision Tree degrades less but still falls to an accuracy of 0.6031 and a recall of 0.5289, with a precision of 1.0000: it raises few ICMP alarms, and those it raises are correct. The Random Forest keeps an AUC of 0.6842 on ICMP, so ranking information survives even here. None of this is visible in the aggregate accuracy of 0.8784.

**Table 4 Protocol-stratified performance of the baselines on NSL-KDD**

| Protocol (n) | Model | Accuracy | Precision | Recall | F1 | AUC |
|---|---|---|---|---|---|---|
| TCP (24,314) | DT | 0.9201 | 0.9194 | 0.9188 | 0.9191 | 0.9507 |
| TCP (24,314) | RF | 0.9272 | 0.9825 | 0.8680 | 0.9217 | 0.9724 |
| UDP (3,518) | DT | 0.9358 | 0.8956 | 0.7519 | 0.8174 | 0.8723 |
| UDP (3,518) | RF | 0.9105 | 0.9163 | 0.5854 | 0.7144 | 0.8876 |
| ICMP (1,872) | DT | 0.6031 | 1.0000 | 0.5289 | 0.6918 | 0.8906 |
| ICMP (1,872) | RF | 0.1848 | 0.9048 | 0.0361 | 0.0695 | 0.6842 |

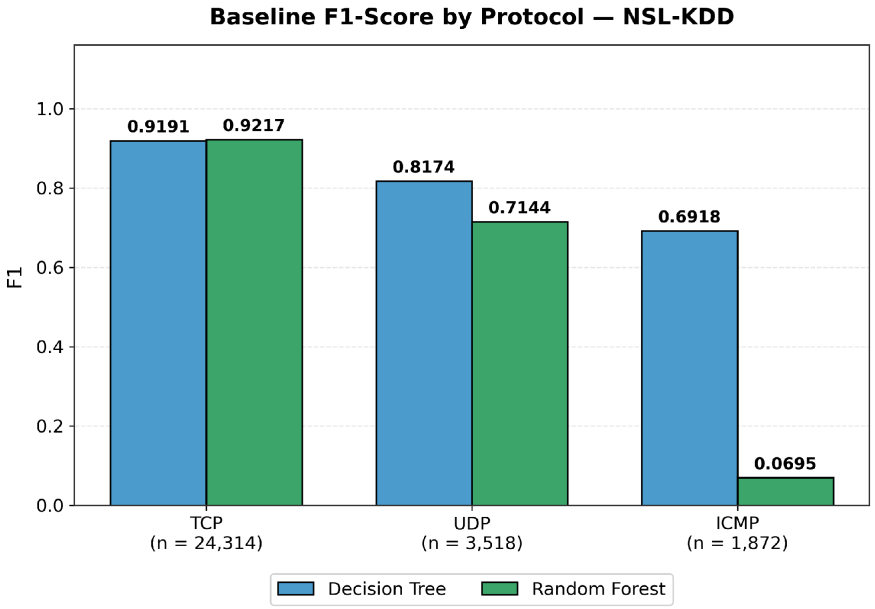



*Fig. 3 F1-score of the two baselines disaggregated by protocol on NSL-KDD. The Random Forest collapses on the attack-dominant ICMP subset*

On UNSW-NB15 the Decision Tree again reaches a recall of 1.0000 on both TCP and UDP through over-prediction (TCP precision 0.6785), while the Random Forest gives a more balanced TCP profile (precision 0.8680, recall 0.7957) and an F1 of 0.9541 on UDP. The six held-out ICMP records of this corpus are all normal, so precision, recall and AUC are undefined there and carry no information.

### *4.2 Convergence of the Linformer*

Fig. 4 shows the training trajectories. On NSL-KDD, training loss fell from 0.1098 to 0.0472 over 20 epochs, a reduction of 57.0%, with most of the fall in the first three epochs. Validation accuracy was 97.03% after one epoch, already above both trained baselines, and ended at 98.17% against a training accuracy of 98.16%. The model reached 95% of its final validation accuracy after one epoch and 99% after three. On UNSW-NB15, training loss fell from 0.2125 to 0.1616, a reduction of 24.0%, and validation accuracy rose from 89.37% to 92.17%, ending 0.32 percentage points above training accuracy; 99% of the final value was reached after eight epochs. In neither corpus did validation loss rise while training loss fell, and validation performance ended marginally above training performance, so there is no sign of overfitting. The terminal loss on UNSW-NB15 is 3.42 times that on NSL-KDD under an identical architecture and procedure, which indicates that the contemporary corpus is the harder problem.

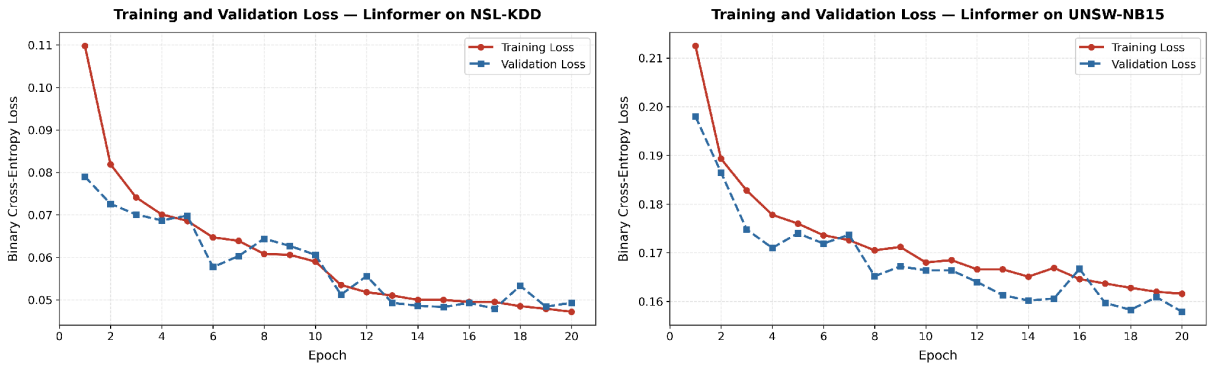



*Fig. 4 Training and validation loss of the Linformer over 20 epochs on NSL-KDD (left) and UNSW-NB15 (right)*

### *4.3 Aggregate performance of the Linformer*

Table 5 reports the held-out results and Fig. 5 the confusion matrices. On NSL-KDD the Linformer reached an accuracy of 0.9823, a precision of 0.9793, a recall of 0.9838 and an AUC of 0.9987. Of 15,450 normal records, 296

raised false alarms (FPR 1.92%); of 14,254 attacks, 231 escaped (FNR 1.62%). The near equality of precision and recall, and of recall and specificity, shows a well-placed operating point, and the MCC of 0.9645 confirms that the result is not an artefact of class prevalence. An AUC of 0.9987 means that for a randomly drawn attack and normal pair, the attack receives the higher anomaly score in 99.87% of cases.

**Table 5 Aggregate performance and error profile of the Linformer**

| Metric | NSL-KDD | UNSW-NB15 |
|---|---|---|
| Accuracy | 0.9823 | 0.9231 |
| Precision | 0.9793 | 0.9326 |
| Recall | 0.9838 | 0.9362 |
| F1-score | 0.9816 | 0.9344 |
| AUC-ROC | 0.9987 | 0.9832 |
| Specificity | 0.9808 | 0.9046 |
| MCC | 0.9645 | 0.8415 |
| TN / FP / FN / TP | 15,154 / 296 / 231 / 14,023 | 16,195 / 1,707 / 1,612 / 23,638 |
| False positive rate | 1.92% | 9.54% |
| False negative rate | 1.62% | 6.38% |
| Balanced accuracy | 98.23% | 92.04% |

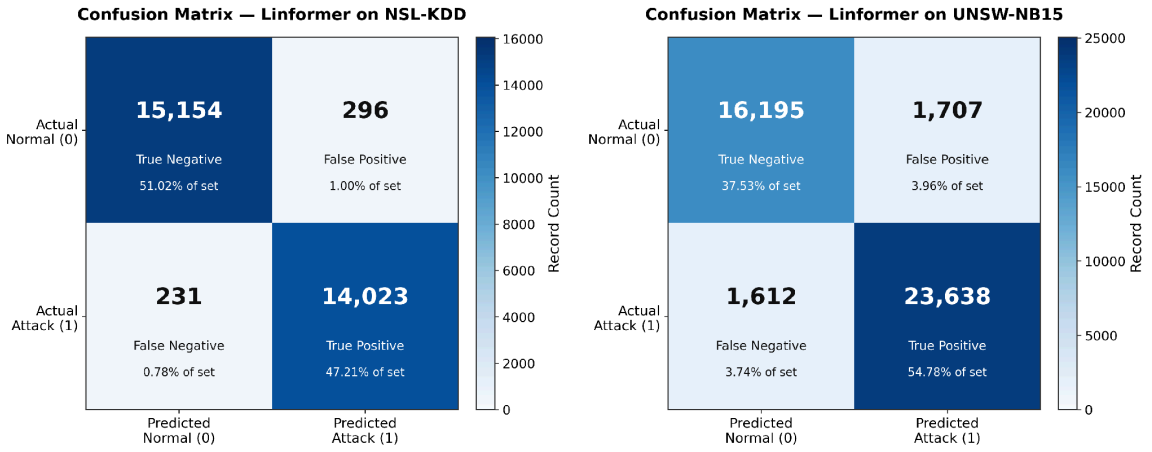



*Fig. 5 Confusion matrices of the Linformer on the NSL-KDD held-out set (left, n = 29,704) and the UNSW-NB15 held-out set (right, n = 43,152)*

On UNSW-NB15 all metrics were lower: accuracy 0.9231, F1 0.9344, AUC 0.9832 and MCC 0.8415. Of 17,902 normal records, 1,707 raised false alarms (FPR 9.54%, 4.97 times the NSL-KDD rate), and of 25,250 attacks, 1,612 escaped (FNR 6.38%, 3.94 times higher). Per 10,000 legitimate flows the detector raises 192 false alarms on NSL-KDD and 954 on UNSW-NB15; per 10,000 attacks it misses 162 and 638. AUC was the most resilient metric, falling only from 0.9987 to 0.9832, so the model continues to rank records well even where its discrete decisions weaken.

Against the best baseline on each metric, the Linformer gains 0.0803 in accuracy, 0.1160 in recall, 0.0869 in F1 and 0.0338 in AUC on NSL-KDD, with precision essentially equal (0.9793 against 0.9795). Expressed as errors removed, it eliminates 81.94% of the misclassifications of the Decision Tree and 85.44% of those of the Random Forest on NSL-KDD, and 49.37% and 33.19% respectively on UNSW-NB15. The Linformer ranks first on nine of the ten metric and corpus combinations; the exception is the degenerate Decision Tree recall of 1.0000 on UNSW-NB15, and the Linformer leads on F1 there by 0.0324.

### *4.4 Protocol-stratified performance of the Linformer*

Table 6 disaggregates the Linformer results for both corpora, and Fig. 6 plots the NSL-KDD case. On NSL-KDD, performance is strong on all three protocols, with every AUC above 0.99. TCP gives the highest MCC (0.9708), with 203 false alarms among 12,310 normal records and 152 missed attacks among 12,004. ICMP gives the highest F1 (0.9921): 1,566 of 1,577 attacks were detected, and a specificity of 0.9525 shows that the model separates the classes instead of leaning on the 84.24% prior, since a flag-everything rule would have a specificity of zero. The model exceeds that trivial rule by 14.42 percentage points in accuracy. The recall of 0.9930 on ICMP compares with 0.0361 for the Random Forest and 0.5289 for the Decision Tree, The protocol-level failure seen in the baselines is gone, which is a larger change than a marginal gain.

**Table 6 Protocol-stratified performance and confusion counts of the Linformer**

| Corpus | Protocol (n) | Attack rate | Precision | Recall | F1 | AUC | FNR | MCC |
|---|---|---|---|---|---|---|---|---|
| NSL-KDD | TCP (24,314) | 49.37% | 0.9832 | 0.9873 | 0.9852 | 0.9991 | 1.27% | 0.9708 |
| NSL-KDD | UDP (3,518) | 19.13% | 0.8845 | 0.8990 | 0.8917 | 0.9900 | 10.10% | 0.8658 |
| NSL-KDD | ICMP (1,872) | 84.24% | 0.9911 | 0.9930 | 0.9921 | 0.9984 | 0.70% | 0.9495 |
| UNSW-NB15 | TCP (24,596) | 45.19% | 0.8923 | 0.8858 | 0.8891 | 0.9711 | 11.42% | 0.7982 |
| UNSW-NB15 | UDP (18,550) | 76.20% | 0.9637 | 0.9757 | 0.9697 | 0.9925 | 2.43% | 0.8704 |
| UNSW-NB15 | ICMP (6) | 0.00% | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined |

All six UNSW-NB15 ICMP held-out records are normal and were classified correctly; the subset supports no inference

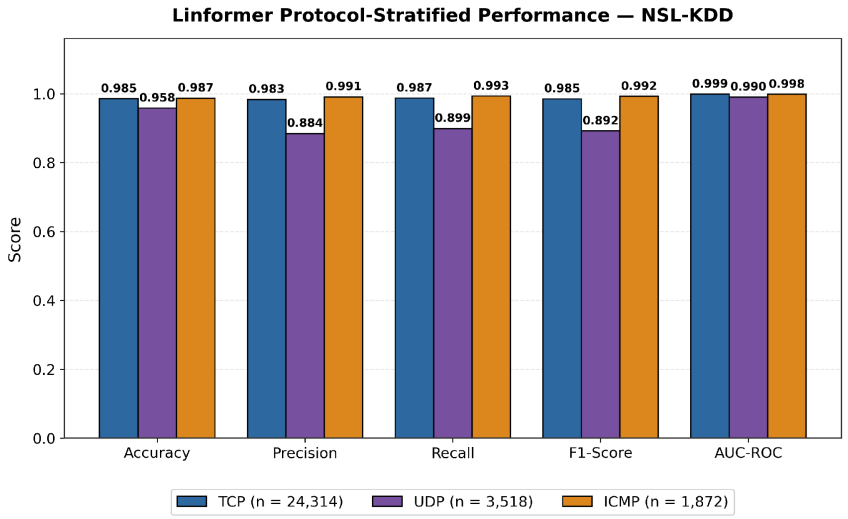



*Fig. 6 Protocol-stratified performance of the Linformer on NSL-KDD*

UDP on NSL-KDD is the weakest subset. Precision falls to 0.8845, MCC to 0.8658 and the FNR rises to 10.10%, more than six times the TCP figure. Only 19.13% of UDP records are attacks: of 684 records flagged, 79 were false alarms, and of 673 attacks, 68 escaped. Because the AUC stays at 0.9900, the model orders UDP records correctly and the loss comes from a threshold that is too high for this subset. On UNSW-NB15 the ordering of the two protocols

reverses. There UDP is attack dominant (76.20%) and gives the stronger result (F1 0.9697, MCC 0.8704), while TCP is close to balanced (45.19%) and gives F1 0.8891 and MCC 0.7982.

### *4.5 Prevalence sensitivity*

The reversal suggests that detection quality follows the attack prevalence of a subset more than the protocol itself. Fig. 7 plots F1 against prevalence for the five assessable subsets. The two most attack-dominant subsets, ICMP on NSL-KDD (84.24%) and UDP on UNSW-NB15 (76.20%), give the two highest F1 values, and the most attack-scarce subset, UDP on NSL-KDD (19.13%), gives the lowest. The Pearson correlation is 0.7773 ($R^2 = 0.6043$), and the fitted slope implies an increase of about 0.0015 in F1 per percentage point of prevalence. Five non-independent points only support a descriptive pattern, not a statistical estimate. The mechanism, though, is plain: a threshold of 0.5 is suited to a prior near one half, and where attacks are scarce it pushes real attacks with middling scores into the normal class.

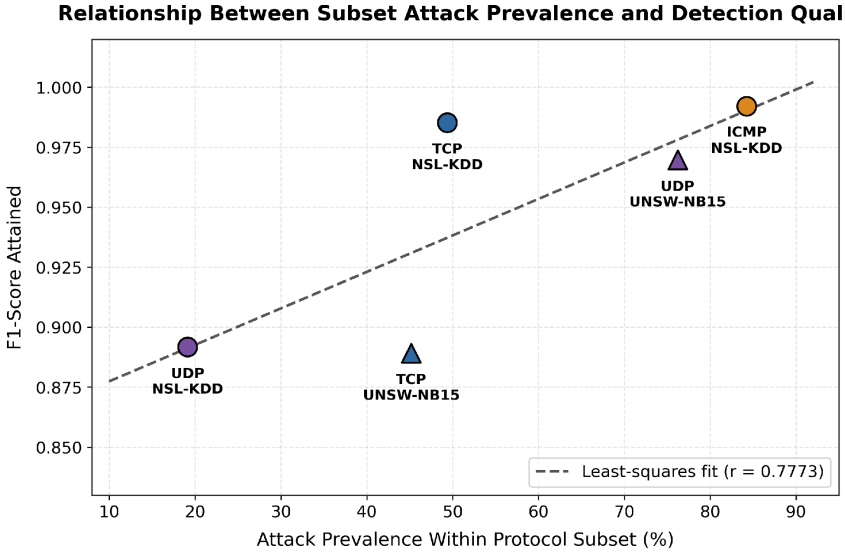



*Fig. 7 F1-score of the Linformer against attack prevalence for the five assessable protocol subsets across both corpora*

### *4.6 Interpretability*

The global importances of the tree baselines on NSL-KDD (Fig. 8) are led by *diff_srv_rate*, the share of recent connections directed at different services. The single tree places about 0.71 of its importance on this attribute and most of the rest on *src_bytes*, while the forest spreads importance over *diff_srv_rate*, *src_bytes*, *same_srv_rate*, *count* and *dst_host_same_srv_rate*. The difference reflects correlated attribute blocks: a single tree picks one representative of a block, while an ensemble samples several. A high *diff_srv_rate* is the fingerprint of port scanning and service enumeration, which is consistent with the 13,758 reconnaissance records (*satan*, *ipsweep*, *portsweep*, *nmap*, *mscan*) in the corpus. On UNSW-NB15 the time-to-live attributes dominate, in keeping with the non-default TTL values emitted by many synthetic attack tools.

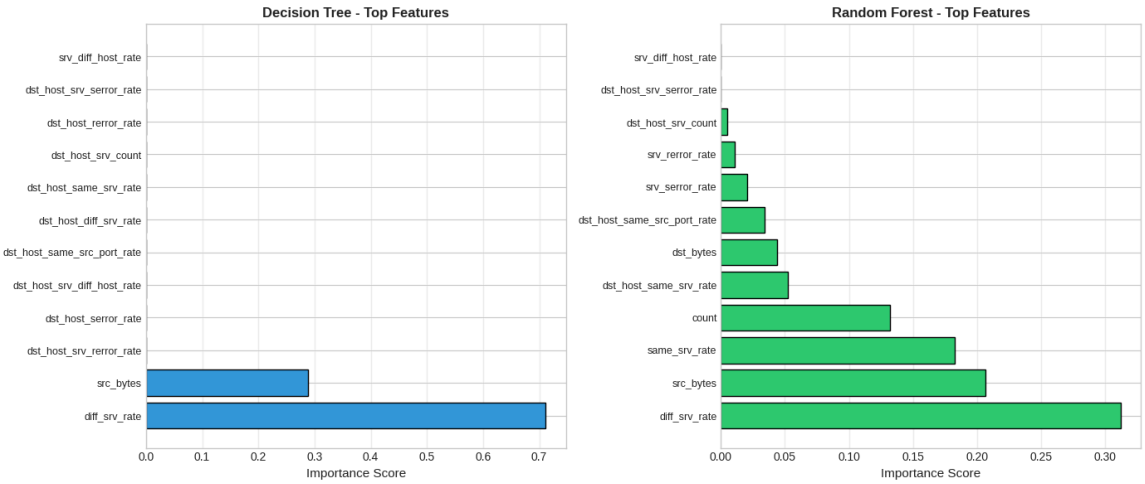



*Fig. 8 Feature importance profiles of the Decision Tree (left) and Random Forest (right) baselines on NSL-KDD*

LIME explanations were produced for one correctly classified record per protocol. On NSL-KDD, the ICMP attack record received an anomaly score of 1.0000 and the TCP and UDP normal records received 0.0000 and 0.0002. The encoded protocol was the largest contributor in all three explanations, followed by the destination host count and the error-rate family, with *wrong_fragment* and *duration* among the ten most influential attributes. On UNSW-NB15, the TCP attack record received a less extreme score of 0.7684, consistent with the higher residual loss on that corpus, and destination TTL together with the protocol identifier dominated the attributions. The agreement between the global tree importances and the local model-agnostic attributions on the same attribute families suggests that the detector has learned real behavioural regularities, not artefacts of the fitting procedure. Because LIME is local and linear, these attributions describe behaviour near the explained record only.

### *4.7 Detection under operational prevalence*

Sensitivity and specificity are properties of the classifier, but precision also depends on prevalence. Holding the measured recall and specificity fixed, Table 7 projects the alerts per million flows as attack prevalence falls. At 50% prevalence the projected NSL-KDD precision is 0.9809, close to the measured 0.9793, which supports the projection. At 5% prevalence it falls to 0.7299, at 1% to 0.3415 and at 0.5% to 0.2051. The UNSW-NB15 model, with its higher FPR, falls to 0.0902 at 1% prevalence (Fig. 9). At 1% prevalence and one million flows per day, the NSL-KDD model would generate 28,805 alerts, 65.85% of them false, and the UNSW-NB15 model 103,761 alerts, 90.98% of them false. At an optimistic 100 reviews per analyst per day, these volumes would require 288 and 1,038 analysts. The models have not deteriorated; this is the arithmetic of rare-event detection (Axelsson 2000).

**Table 7 Projected detection statistics per million flows at different attack prevalences**

| Prevalence | NSL-KDD alerts | NSL-KDD FP | NSL-KDD precision | UNSW-NB15 alerts | UNSW-NB 15 FP | UNSW-NB15 precision |
|---|---|---|---|---|---|---|
| 0.5% | 23,982 | 19,063 | 0.2051 | 99,557 | 94,876 | 0.0470 |
| 1.0% | 28,805 | 18,967 | 0.3415 | 103,761 | 94,399 | 0.0902 |
| 5.0% | 67,390 | 18,201 | 0.7299 | 137,393 | 90,585 | 0.3407 |
| 10.0% | 115,622 | 17,243 | 0.8509 | 179,433 | 85,817 | 0.5217 |
| 25.0% | 260,317 | 14,369 | 0.9448 | 305,554 | 71,514 | 0.7660 |
| 50.0% | 501,476 | 9,579 | 0.9809 | 515,755 | 47,676 | 0.9076 |

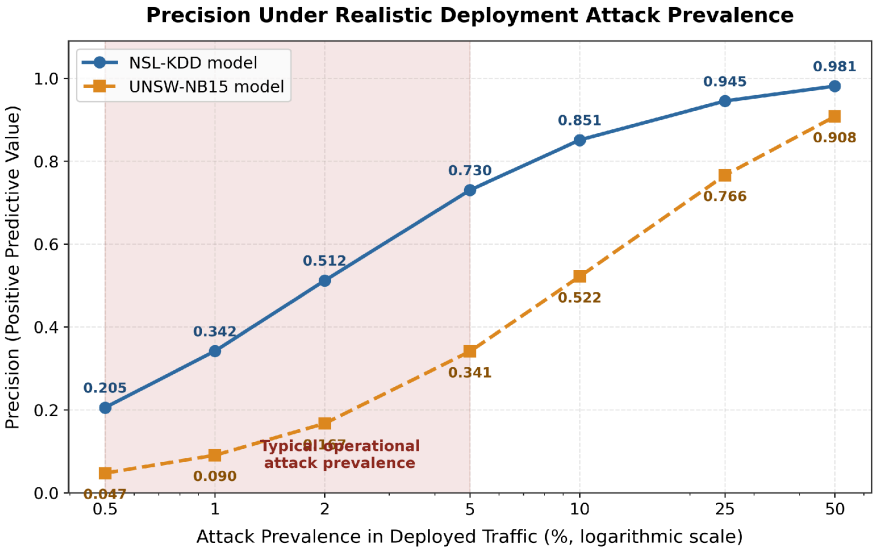



*Fig. 9 Projected precision of the Linformer detectors as attack prevalence falls toward operational levels*

### *4.8 Comparison with published results*

Table 8 places the Linformer against the methods tabulated by Jeong et al. (2024), and Fig. 10 plots the NSL-KDD accuracies. The Linformer has the highest accuracy of the eight methods on NSL-KDD (0.9823 against 0.8939 for FPCD-DBN) and of the seven on UNSW-NB15 (0.9231 against 0.8662). Against FPCD-DBN it gains 0.0844 in F1 and 0.0350 in AUC on NSL-KDD, and 0.0516 and 0.0186 on UNSW-NB15, removing 83.32% and 42.53% of that model's misclassifications. The AUC comparison is against the strongest of the four architectures reported by Jeong et al. (2024), whose values were 0.9637 and 0.9646. Several prior methods show one-sided profiles, such as the denoising autoencoder of Aygun and Yavuz (2017) with precision 0.9648 and recall 0.8308, or the SVM of Malaiya et al. (2019) with precision 0.9730 and recall 0.7560, whereas the Linformer is balanced across both. These margins carry a caveat about evaluation protocol, taken up in Section 5. Newer studies report higher accuracies on the same corpora, for example above 99% (Xi et al. 2024) and 97.93% on UNSW-NB15 (Farhan et al. 2025), but under different splits and feature selection. Table 8 is therefore limited to the methods Jeong et al. (2024) collected under one protocol.

**Table 8 Comparison with methods reported in the literature on the two corpora**

| Corpus | Method | Source | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|---|---|
| NSL-KDD | DNN | Tang et al. (2016) | 0.7575 | NR | NR | NR |
| NSL-KDD | RNN-IDS | Yin et al. (2017) | 0.8129 | NR | NR | NR |
| NSL-KDD | RT | Aggarwal and Sharma (2015) | 0.8621 | 0.8078 | 0.9416 | 0.8696 |
| NSL-KDD | AE | Aygun and Yavuz (2017) | 0.8828 | 0.9123 | 0.8786 | 0.8951 |
| NSL-KDD | DAE | Aygun and Yavuz (2017) | 0.8865 | 0.9648 | 0.8308 | 0.8928 |
| NSL-KDD | LSTM | Naseer et al. (2018) | 0.8900 | NR | NR | NR |
| NSL-KDD | FPCD-DBN | Jeong et al. (2024) | 0.8939 | 0.8951 | 0.8983 | 0.8972 |
| NSL-KDD | **Linformer** | **This study** | **0.9823** | **0.9793** | **0.9838** | **0.9816** |
| UNSW-NB15 | NB | Khan and Gumaei (2019) | 0.7639 | 0.7820 | 0.7640 | 0.7729 |
| UNSW-NB15 | SVM | Malaiya et al. (2019) | 0.8190 | 0.9730 | 0.7560 | 0.8210 |

| | | | | | | |
|---|---|---|---|---|---|---|
| UNSW-NB 15 | RF | Zhou et al. (2020) | 0.8363 | 0.8690 | 0.8360 | 0.8522 |
| UNSW-NB 15 | MemAE | Min et al. (2021) | 0.8530 | 0.8774 | 0.8530 | 0.8526 |
| UNSW-NB 15 | FPCD-DBN | Jeong et al. (2024) | 0.8662 | 0.8841 | 0.8816 | 0.8828 |
| UNSW-NB 15 | **Linformer** | **This study** | **0.9231** | **0.9326** | **0.9362** | **0.9344** |

Prior results as tabulated by Jeong et al. (2024); NR not reported

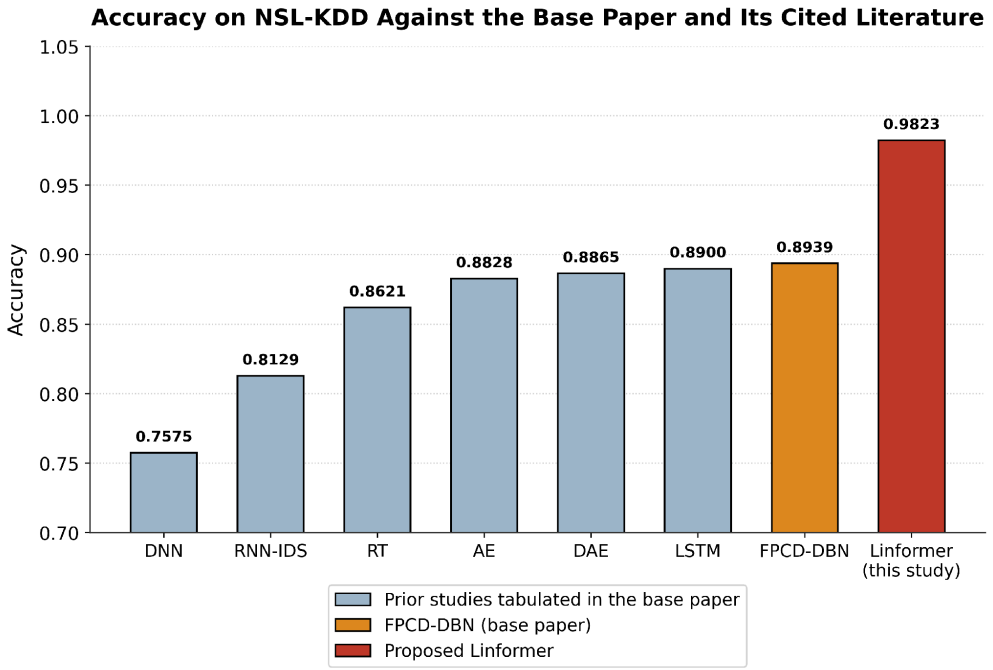



*Fig. 10 Accuracy on NSL-KDD of the Linformer against the base study and the literature it cites*

## 5. DISCUSSION

The key inference of this work is not really about architecture but about evaluation. Only reported in aggregate, the Random Forest would be considered an average quality detector with an accuracy approaching that of 0.88. When used on protocols stratified, it can handle TCP and UDP, as well as being virtually useless when used on ICMP, where flooding and reconnaissance is a literal worry. The failure at the protocol level can remain unknown with the use of aggregate reporting and a detector that should be deployed is to be rated protocol-by-protocol. The same analysis reasons as to why they fail: the three protocols are characterised by different class priors, and a globally optimised threshold is locally incorrect wherever the prior is different to the global one.

This is made into a practical suggestion by the commonality trend of Section 4.5. All the assessable subsets maintained an AUC of 0.97, and 0.9900 even on the weakest, indicating that the ranking would require a more stringent threshold is already included in the anomaly score. Retraining should only restore a significant portion of the missed attacks without training new thresholds on the validation partition, which can be done by calibrating each protocol separately on the validation partition, with lower thresholds on attack-sparse subsets of the protocol like UDP on NSL-KDD. There is minimal cost involved and it can be tested on the existing models.

The difference between the two corpora is a story by itself. False alarms grew by a factor of 4.97 and missed attacks by 3.94 on UNSW-NB15. Three factors are probable: its attack types, like fuzzers, are skewed to look like legitimate traffic; its attack-dependent prior precludes the best total threshold of 0.5; and its flow-based features do not have the aggregate rate data that were so discriminative on NSL-KDD. The FPCD-DBN of Jeong et al. (2024) has also lower

scores in UNSW-NB15 and the difference between the two models has diminished there, which points to the fact that it is with data that the difficulty occurs. The NSL-KDD values need to be used as an upper limit and the practitioner would expect to approach the UNSW-NB15 values on a newer network.

Section 4.7 projections indicate the manner in which the detector needs to be deployed. The automatic block of 19,000 valid flows per million would still be interrupted at an automatic blocking of 1.92. The detector would be more preferable as a triage phase ranking and describing alerts to a second analysis level, and the FPR, rather than the accuracy, should inform model choice; at which the NSL-KDD-trained model scores better by about 5 times, which is not disclosed by the 0.9823 or 0.9231 accuracies. This role is justified by the LIME explanations, which present an alert in the form of named traffic attributes that can be acted on by an administrator.

### *5.1 Limitations*

These claims were circumscribed by a number of limitations. To start with, the baselines to compare it with were intentionally small: a depth-2 tree and five depth-2 trees, about 7 and 35 learnable parameters, respectively, versus 1,294,084 of the Linformer. The margins indicate that the Linformatics beats tiny unreadable baselines. They fail to demonstrate that attention is any better than tree ensembles; a reasonable experiment would permit the depths of 10 to 15 and about 100 estimators. There is also a difference between the two baselines in terms of class weighting, thus making the comparison between them confound the ensembling and reweighting. Second, the threshold of each model was not tuned and such tuning was likely to benefit the baselines more than the Linformer. Third, this work combined the distributed training and testing files and randomly selected an 80/20 split, but Jeong et al. (2024) utilized the official splits (22,544 and 82,332 held-out records). Signatures not included in the training are included in the official NSL-KDD test set, thus some of the NSL-KDD margin over FPCD-DBN is due to a simpler protocol, and its size in the official split is not determined. Fourth, the experiments were only repeated with a fixed seed, thus the confidence intervals are not provided; the reason behind the 610 retained NSL-KDD duplicates could equally contribute a small optimistic bias. Fifth, UNSW-NB15 does not possess any usable ICMP evidence, and CIC-IDS-2017 has not been tested. Lastly, an automated response layer, streaming pipeline or end-to-end latency measurement was not constructed and thus the work confirms the presence of the detection layer of a response-capable system, rather than a closed-loop system.

## 6. CONCLUSION

This paper has introduced a Linformer-based TCP, UDP and ICMP anomaly detector and evaluated it on a protocol-stratified protocol on both NSL-KDD and UNSW-NB15. On NSL-KDD, the detector achieved a top F1 of 0.9816 and AUC of 0.9987, and UNSW-NB15 a top F1 of 0.9344 and AUC of 0.9832, the best performing of the three implemented models and outperforming the published methods compared, and did not overfit on either corpus. Stratification revealed that aggregate measures concealed a Random Forest ICMP recall of 0.0361, which the Linformer increased to 0.9930 and that detection quality corresponded to the prevalence of each protocol subset in the attacks. Predicted to be operational prevalence, benchmark accuracy of 0.9793 is 0.3415 at 1% attack traffic,) hence supporting the idea of implementing such detectors as triage tools. A re-run of the detector on the official NSL-KDD and UNSW-NB15 partitions, and on CIC-IDS-2017; comparison with adequately powered tree ensembles and a production signature engine, like Snort or Suricata; calibration of per-protocol thresholds on validation data; other repeat all the experiments across multiple seeds to report confidence intervals; and construct and test the automated response layer, including streaming inference, and integration with firewall or intrusion prevention infrastructure.

## REFERENCES

Aggarwal P, Sharma SK (2015) Analysis of KDD dataset attributes: class wise for intrusion detection. Procedia Comput Sci 57:842–851

Alkhater N (2026) A rigorous comparative study of supervised machine learning techniques for network anomaly detection: empirical insights from the UNSW-NB15 dataset. Computers 15(5):285. https://doi.org/10.3390/computers15050285

Axelsson S (2000) The base-rate fallacy and the difficulty of intrusion detection. ACM Trans Inf Syst Secur 3(3):186–205

Aygun RC, Yavuz AG (2017) Network anomaly detection with stochastically improved autoencoder based models. In: 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud). IEEE, pp 193–198

Ba JL, Kiros JR, Hinton GE (2016) Layer normalization. arXiv preprint arXiv:1607.06450

Bace R, Mell P (2001) Intrusion detection systems. NIST Special Publication 800-31. National Institute of Standards and Technology, Gaithersburg

Berglund M, Raiko T (2013) Stochastic gradient estimate variance in contrastive divergence and persistent contrastive divergence. arXiv preprint arXiv:1312.6002

Breiman L (2001) Random forests. Mach Learn 45(1):5–32

Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) Classification and regression trees. Wadsworth and Brooks, Monterey

Buczak AL, Guven E (2016) A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Commun Surv Tutor 18(2):1153–1176

Chicco D, Jurman G (2020) The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics 21(1):6

Chinnasamy R, Subramanian M, Veerappampalayam Easwaramoorthy S, Cho J (2025) Deep learning-driven methods for network-based intrusion detection systems: a systematic review. ICT Express 11(1):181–215. https://doi.org/10.1016/j.icte.2025.01.005

Denning DE (1987) An intrusion-detection model. IEEE Trans Softw Eng SE-13(2):222–232

Farhan M, Waheed ud din H, Ullah S et al (2025) Network-based intrusion detection using deep learning technique. Sci Rep 15:25550. https://doi.org/10.1038/s41598-025-08770-0

Fawcett T (2006) An introduction to ROC analysis. Pattern Recognit Lett 27(8):861–874

Fischer A, Igel C (2012) An introduction to restricted Boltzmann machines. In: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. Springer, Berlin, pp 14–36

Hendrycks D, Gimpel K (2016) Gaussian error linear units (GELUs). arXiv preprint arXiv:1606.08415

Hinton GE (2009) Deep belief networks. Scholarpedia 4(5):5947

Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on Machine Learning, pp 448–456

Jeong J, Park S, Lim J, Kang J, Shin D, Shin D (2024) A study on network anomaly detection using fast persistent contrastive divergence. Symmetry 16(9):1220. https://doi.org/10.3390/sym16091220

Khan FA, Gumaei A (2019) A comparative study of machine learning classifiers for network intrusion detection. In: Artificial Intelligence and Security: 5th International Conference, ICAIS 2019. Springer, pp 75–86

Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations

Liao HJ, Lin CHR, Lin YC, Tung KY (2013) Intrusion detection system: a comprehensive review. J Netw Comput Appl 36(1):16–24

Malaiya RK, Kwon D, Suh SC, Kim H, Kim I, Kim J (2019) An empirical evaluation of deep learning for network anomaly detection. IEEE Access 7:140806–140817

Min B, Yoo J, Kim S, Shin D, Shin D (2021) Network anomaly detection using memory-augmented deep autoencoder. IEEE Access 9:104695–104706

Moustafa N, Slay J (2015) UNSW-NB15: a comprehensive data set for network intrusion detection systems. In: 2015 Military Communications and Information Systems Conference (MilCIS). IEEE, pp 1–6

Moustafa N, Slay J (2016) The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Inf Secur J Glob Perspect 25(1–3):18–31

Naseer S, Saleem Y, Khalid S, Bashir MK, Han J, Iqbal MM, Han K (2018) Enhanced network anomaly detection based on deep neural networks. IEEE Access 6:48231–48246

Pang G, Shen C, Cao L, van den Hengel A (2021) Deep learning for anomaly detection: a review. ACM Comput Surv 54(2):38

Paszke A, Gross S, Massa F et al (2019) PyTorch: an imperative style, high-performance deep learning library. Adv Neural Inf Process Syst 32:8024–8035

Pedregosa F, Varoquaux G, Gramfort A et al (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830

Pinto Neto EC, Iqbal S, Buffett S et al (2025) Deep learning for intrusion detection in emerging technologies: a comprehensive survey and new perspectives. Artif Intell Rev 58(11):340. https://doi.org/10.1007/s10462-025-11346-z

Powers DMW (2011) Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. J Mach Learn Technol 2(1):37–63

Ribeiro MT, Singh S, Guestrin C (2016) "Why should I trust you?": explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp 1135–1144

Ring M, Wunderlich S, Scheuring D, Landes D, Hotho A (2019) A survey of network-based intrusion detection data sets. Comput Secur 86:147–167

Sommer R, Paxson V (2010) Outside the closed world: on using machine learning for network intrusion detection. In: Proceedings of the IEEE Symposium on Security and Privacy, pp 305–316

Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958

Tang TA, Mhamdi L, McLernon D, Zaidi SAR, Ghogho M (2016) Deep learning approach for network intrusion detection in software defined networking. In: 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM). IEEE, pp 258–263

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

Tieleman T, Hinton G (2009) Using fast weights to improve persistent contrastive divergence. In: Proceedings of the 26th Annual International Conference on Machine Learning. ACM, pp 1033–1040

Umer M, Tahir M, Sardaraz M et al (2025) Network intrusion detection model using wrapper based feature selection and multi head attention transformers. Sci Rep 15:28718. https://doi.org/10.1038/s41598-025-11348-5

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. Adv Neural Inf Process Syst 30:5998–6008

Wang S, Li BZ, Khabsa M, Fang H, Ma H (2020) Linformer: self-attention with linear complexity. arXiv preprint arXiv:2006.04768

Xi C, Wang H, Wang X (2024) A novel multi-scale network intrusion detection model with transformer. Sci Rep 14:23239. https://doi.org/10.1038/s41598-024-74214-w

Xu J, Wu H, Wang J, Long M (2022) Anomaly Transformer: time series anomaly detection with association discrepancy. In: Proceedings of the International Conference on Learning Representations

Yin C, Zhu Y, Fei J, He X (2017) A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access 5:21954–21961

Zhou X, Hu Y, Liang W, Ma J, Jin Q (2020) Variational LSTM enhanced anomaly detection for industrial big data. IEEE Trans Ind Inform 17:3469–3477