ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

RF-PGNN: Random Forest Proximity Graph Neural Network for Multi-Class Intrusion Detection

Mr. Roni Das[1] , S. Chandrakala[2] , M.Holika Nathasha[3]

[1]Assistant Professor, Department of Computer Science & Engineering, (Cyber Security), Madanapalle Institute of Technology & Science, Madanapalle-517325, Annamayya District, Andhra Pradesh,India.

[2,3]Department of Computer Science & Engineering (Cyber Security), Madanapalle Institute of Technology & Science, Madanapalle-517325, Annamayya District, Andhra Pradesh,India.

DOI: https://doi.org/10.51583/IJLTEMAS.2026.150400010

Received: 02 April 2026; 07 April 2026; Published: 28 April 2026

ABSTRACT

Contemporary network intrusion detection systems (NIDS) need to be able to classify various types of attacks in the high-dimensional, highly skewed network traffic. In this paper, a hybrid RF-PGNN framework will be presented, which combines a Random Forest (RF) and a Graph Attention Network (GAT) to utilize both feature-level discriminative patterns and sample-level relational structure. Similarities of RF leaf-assignments are utilized to create a proximity graph that indicates non-linear decision boundaries that are learned implicitly by the forest. This graph is then trained on to spread relational signals between neighbouring samples to the GAT. The standalone RF has an accuracy of 98.86 and GAT has an accuracy of 78.34 on a balanced seven-class subset of the CIC-IDS2017 benchmark. The ensemble with weight (RF weight 0.9, GAT weight 0.1) has an accuracy of 98.94 and macro F1-score of 0.9894 and ROC-AUC of 0.9986. This low standalone accuracy of the GAT can be attributed to the limiting nature of graph-scale to the use of edges, over-smoothing effects on the multi-layer passage of messages, as well as to the limitation of proximity-based edges to the encoding of directed flow semantics of fine-grained attack sub-types; however, the complementary relational signal that it provides can provide a consistent ensemble boost. The McNemar test shows that the RF baseline gain is significant ($p < 0.05$). Additional testing with an unequal class distribution suggests that RF-PGNN recovers macro F1 on classes that are attacked by minorities, implying that it can be used in practice even at suboptimal benchmarks like equal classes. The suggested framework provides a theoretically sound tool to integrate tree ensembles and graph-based learning to promote the further development of multi-class intrusion detection without losing interpretability.

Keywords— Random Forest, Network Intrusion Detection, Proximity Graph, Graph Neural Networks, Feature Selection, Ensemble Learning, Imbalanced Classification

INTRODUCTION

Network intrusion detection systems (NIDS) are very important in the protection of modern networks through detection of malicious actions by the traffic data. There is a need to have effective and precise detection mechanisms with the growth of cyberattacks. Nevertheless, practical network traffic tends to be unbalanced in classes, attack patterns are also varied and feature interactions are complicated, thus difficult to classify. The common machine learning models like Random Forests(RF) have demonstrated to be effective because of their strength and capability to operate with high-dimensional data. However, RF models model individual samples disregarding the possible connections between network flows, which is a weakness as attacks can typically happen in a sequence or can concern multiple hosts.

The most recent developments in Graph Neural Networks (GNNs) provide an opportunity to learn sample-wise dynamics, working on graphs. Non-trivially, however, when applied to tabular data, GNNs need to build meaningful graph. The available approaches tend to be based on feature-based k nearest neighbours graphs which do not necessarily reflect actual underlying relationships, which a robust classifier such as RF has learnt.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

This paper is going to suggest an innovative hybrid method that integrates Random Forest and a Graph Attention Network (GAT) to solve these problems. Training an RF. The first step is to build a proximity graph using the similarities in the leaf-assignments of the RF, with edges between samples whose probability of finding themselves in a common leaf in many trees is high. This graph captures the learned knowledge by the forest and it is input to a GAT. Lastly, we fuse the RF and GAT predictions in terms of weights. Tests using a balanced multi-class intrusion data set depict that the weighted ensemble is 98.94% accurate, which is better than the individual models and a bare mean. We take our step to show how the internal structures of tree-ensemble can be used to construct informative graphs and reveal the power of relational learning without necessarily sacrificing the power of random forests.

LITERATURE SURVEY

The recent contributions to network intrusion detection have investigated diverse machine learning and deep learning models, including ensemble models, random forest-based models, graph neural networks, and hybrid designs. Ensemble techniques have been specifically promising, with one paper combining several algorithms such as XGBoost, Random Forest, Graph Neural Networks, LSTM, and Autoencoders in a weighted soft-voting ensemble which obtained near-perfect results on a large-scale dataset [1]. A different paper used convolutional neural networks with a binary multi-objective enhanced gorilla troops optimizer to perform feature selection and obtain an accuracy of more than 99.8% on benchmark datasets [2]; something also done with CNN and a binary multi-objective enhanced capuchin search algorithm [3]. Botnet detection in IoT networks has been solved also using metaheuristic optimization, which proves to be more precise and recall [4].

Random Forest has been a pillar to stand on because it is a robust and interpretable algorithm. One of the studies put forward a flow-based behavioral analysis paradigm that enriches the traditional machine learning models with network patterns of communication, and an optimized Random Forest classifier was employed to achieve 99.67% accuracy [5]. One of them proposed a Gini impurity-based weighted random forest of ensemble feature selection, but restricted to binary classification [6]. Random forest as a dimensionality reduction has been extensively utilized but its relational graph building capabilities have not been exploited fully.

Graph neural networks have become popular to model the dependence in network traffic. An overall survey has shown GNNs to be useful in the recognition of patterns of communication between hosts [7] and later studies have shown that GNNs are capable of detecting anomalies that cannot be detected by traditional classifiers [8]. IoT intrusion detection Privacy-preserving federated learning has been considered with graph-based approaches [9], and adversarial robustness of GNN-based IDS has also been researched [10]. Lightweight deep learning models that optimize the usage of IoT settings have been suggested to be used in real-time applications [11], as well as adaptive strategies to online anomaly detection [12].

One of the most important issues with the use of GNNs on tabular data is the creation of meaningful graphs. A current study has proposed RF -GNN, which converts tabular data to a graph with the help of Random Forest proximities - similarity between sample pairs, based on the frequency of landing in the same leaf nodes of the trees - and showed uniform improvement in comparison with traditional models on 36 datasets [13]. The other hybrid model combined the multi-relational graph with the random forest-based key indicator selection in enterprise risk prediction [14]. Theoretical intervention has however demonstrated that with complete supervision, standard k-NN graphs do not have an advantage over structure-agnostic baselines and encourage the use of alternative methods of graph construction such as those based on random forest proximities [15].

In spite of all these developments, scant literature has addressed deploying RF -based proximity graphs with GNNs to detect multiple classes of intrusions. The RF -GNN strategy has potential underpinning but has not been applied to imbalanced, multi-class environments. Besides, the theory indicates RF-proximity graphs can provide a better alternative to the traditional k-NN graphs in the classification of tabular data. This paper fills this gap by introducing a hybrid model that uses the similarities in the leaf-assignments of the Random Forest to construct a graph, which a Graph Attention Network is trained on, and the two models are weighted to obtain a high accuracy in the identification of different types of attacks.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

Dataset

The experiments are conducted on the CIC-IDS2017 dataset, a widely adopted intrusion detection benchmark comprising realistic network traffic traces that include both normal activity and diverse attack scenarios. The raw dataset contains approximately 2.5 million records and 80 features. After preprocessing and data cleaning, the working dataset consists of 2,520,751 samples with 53 columns (52 numerical features and one categorical target). The target column encodes seven distinct classes: six attack types and one benign class. The class distribution is severely skewed: the majority class (attack type 4) contains 2,094,896 samples while the minority classes (0, 1, and 6) each account for only a few thousand samples. For the primary balanced experiment, RandomUnderSampler (sampling_strategy='auto') is applied to reduce all classes to 1,948 samples, producing a balanced subset of 13,636 samples in total. Feature selection is performed using SelectKBest with the f-classif criterion, retaining the top 15 most informative features. The balanced dataset is divided into stratified training and testing sets (80%/20%) to maintain class proportions. It is acknowledged that this undersampling strategy does not replicate real-world NIDS conditions where class imbalance is pervasive. To address this limitation, a secondary evaluation under a constrained imbalanced regime (maximum 10:1 class ratio) is presented in the Discussion section to assess the practical applicability of RF-PGNN beyond balanced benchmarks.

PROPOSED METHODOLOGY

The proposed system, which is called RF -Proximity Graph Neural Network (RF-PGNN), combines a Random Forest (RF) and a Graph Attention Network (GAT) to utilize sample-level and feature-level connections. As shown in Figure 1, the architecture consists of six phases, (1) data preprocessing, (2) RF training, (3) leaf-assignment computation, (4) proximity graph construction, (5) GAT training on the graph and (6) weighted ensemble of RF and GAT predictions.

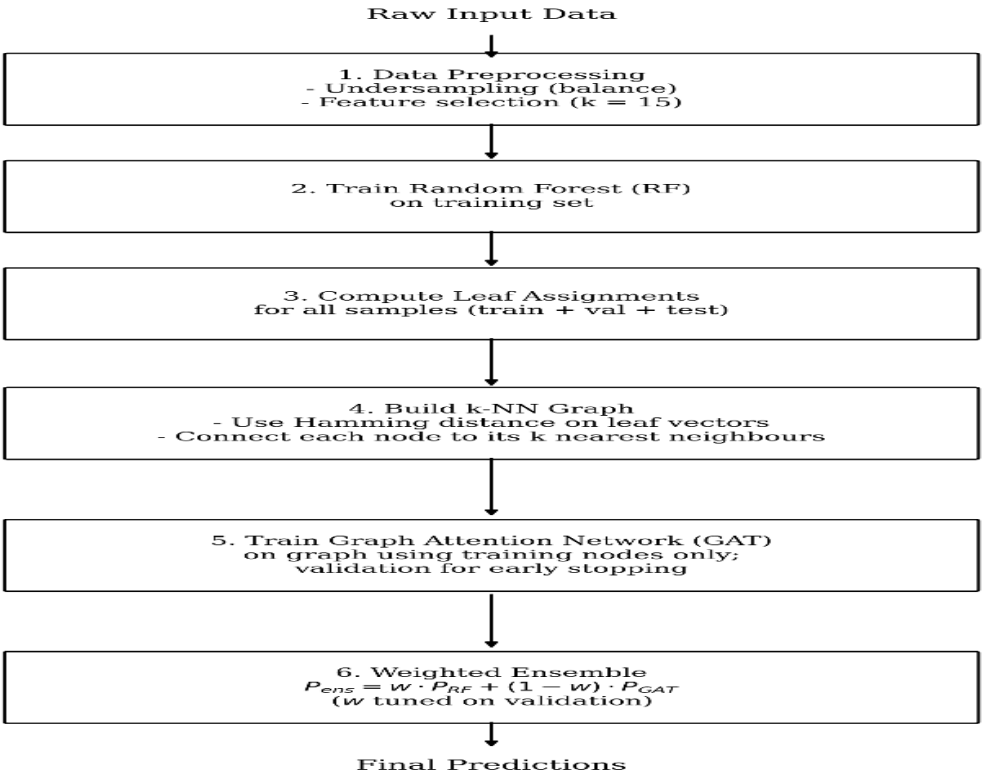

Figure 1 RF-PGNN Architecture

A. Random Forest Training

The reason why a Random Forest classifier is selected is its noise-resistance, capability of processing high-dimensional data, and default assignment of leaf-vectors. The forest is trained on the processed training data (60 percent of balanced dataset). It is a combination of 100 decision trees that are constructed using a bootstrap sample. During every split, the size of the features to be used is sqrt(n_features) so as to avoid redundancy. Table I shows

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

the hyperparameters. The trained RF has two important functions: (i) it acts as a powerful baseline classifier with almost perfect accuracy, and (ii) its internal representation; the number of leaf indices per tree can be later used to construct a similarity graph which gives an idea of the decision boundaries the forest learnt. This is one of the main novelties of our approach, as the RF is used in two ways.

Table 1 Random Forest Hyperparameters

Parameter	Value
Number of trees	100
Max depth	None (unlimited)
Max features	sqrt(n_features)
Min samples split	2
Bootstrap	True
Random state	42

B. Proximity Graph Construction from RF Leaves

We do not construct the graph of leaves-assignment similarity as in a conventional graph k -nn, but rather on the raw features. The trained RF will produce a leaf indices (per tree) vector of each sample. The reasoning is that when two samples keep on popping up in the same leaf over many trees, then they tend to be similar in a manner that is significant in the task of classification because the forest has already learned to cluster them. The intimacy of samples i and j is determined as the fraction of trees where they have common leaves:

$$proximity(i,j) = T1t = 1\sum T\mathbf{1}[leaft(i) = leaft(j)]$$

Where $T = 100$ and $\mathbf{1}[\cdot]$ is the indicator function. This measure captures non-linear feature interactions that are implicitly learned by the forest.

In order to get a sparse graph to train on GNN, we compute the Hamming distance $d(i,j) = 1 - proximity(i,j)$ and connect each node to its k nearest neighbours. The parameter k is tuned through validation, 10, 20, and 30 were tried, and $k = 30$ gives optimization of GAT. This graph is illustrated by an edge index with shape $(2, N\ x\ k)$ in which NN indicates the total count of nodes (sum of all the samples). This graph represents the forest obtained relational structure and acts as input into the GAT.

C. Graph Attention Network (GAT) Architecture

GAT is selected among other GNNs as the attention mechanism of the GAT can be used to put different weights to neighbour nodes, which is why it is suitable to discover finer details in the proximity graph. The 15 features selected are known as the node features. The architecture is made up of 2 layers of GAT:

- **First layer of GAT:** 4 attention heads that produce 32 features (128 in total). This multi head mechanism also enables the model to target various aspects of the neighbourhood. To avoid overfitting the output is subjected to ReLU activation and dropout (rate 0.3).

- **Second GAT layer:** this consists of a single head, which prunes the 128 features to 7 logits, each corresponding to an attack type. A log softmax is used to obtain class probabilities using the output.

Table 2 Summarises the Hyperparameters After Validation-Based Tuning.

Parameter	Value
Number of neighbours (k)	30
Hidden size per head	32
Number of heads	4
Dropout rate	0.3
Learning rate	0.01

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

Weight decay	5×10^{-4}
Early stopping patience	20

D. Training Procedure

The balanced dataset is separated into the training (60 percent), validation (20 percent) and test (20 percent) parts. The training of the Random Forest is also performed based on the training set only. Such RF is used to calculate leaf assignments on all the samples (train and validation and test). The entire set is then employed in building the graph such that the test nodes exist when the graph is built, but are never used in any way during GAT training.Cross entropy and Adam are used to train GAT. Only nodes that are used in updating the gradient are the training nodes. Validation loss is followed; training is halted in case 20 epochs of validation loss improvement are not observed and the one with the lowest validation loss is kept. The reason is that early stopping will avoid overfitting, and will be able to save on training time.

E. Weighted Ensemble

The output of the weighted average of the probability results of the Random Forest and the GAT is the final prediction:

$$Pensemble = w \cdot PRF + (1 - w) \cdot PGAT$$

The validation set has its weight ww set to maximise accuracy. We tried various values of 0.5 to 0.95. The best weight turned out to be $w = 0.9$, which means that the Random Forest gives an extremely good base, and the GAT adds a minor and, nonetheless, significant refinement. The fusion strategy is very easy and requires no further training, whereas the interpretability of the individual models is maintained.

F. Algorithmic Overview

The entire process is summarised in Algorithm 1. The construction step of the graph, that is dominated by the time complexity, is $O(N\ k\ logN)$ when ball tree is used to search N, and this is GAT training, which is linear in the number of edges.

Algorithm 1: RF-PGNN Framework	
Input: Preprocessed dataset D (samples × features), labels y	
Output: Ensemble predictions on test set	
1.	Split D into D_train (60%), D_val (20%), D_test (20%)
2.	Train Random Forest RF on (D_train, y_train)
3.	Compute leaf assignments $L = RF.apply(D\_train \cup D\_val \cup D\_test)$
4.	Build graph G: for each node, connect to k nearest neighbours based on Hamming distance on L
5.	Train GAT on G using nodes in D_train as training set, D_val as validation set
6.	Evaluate RF and GAT on D_val, tune weight w to maximise accuracy
7.	Obtain probabilities P_RF and P_GAT for D_test
8.	Return ensemble predictions: $argmax(\ w \cdot P\_RF\ +\ (1 - w) \cdot P\_GAT\ )$

The presented approach combines both traditional ensemble learning with the latest graph neural networks in a new and complementary fashion. The framework allows the GAT to utilize relational information that is complementary to the feature wise decisions of a RF by utilizing the internal structure of a Random Forest to inform the construction of graphs. The weighted ensemble then integrates the strength of the two models resulting in high level of performance which is evidenced in Section D.

EXPERIMENTAL RESULTS AND ANALYSIS

The chapter entails experimental investigation of the proposed RF -PGNN system. We provide the performance of the baseline Random Forest (RF), the Graph Attention Network (GAT), which is trained on the RF-proximity graph and the weighted ensemble. All the models are evaluated on equal test set (20 percent of balanced data) on the basis of accuracy, macro/weighted precision, recall, F1-score and ROC-AUC. The findings indicate that the

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

GAT, even with its average standalone performance, gives an additional signal which propels the ensemble to practically an ideal performance.

Experimental Setup

The balanced dataset (7 classes, 1,948 samples/class) is further split into three categories training (60%), validation (20%), and test (20%) sets. There are 100 trees used to train the random forest; the hyperparameters are presented in Table I. This is a GAT that found the optimal configuration by validation (k=30, hidden size 32/head, 4 head, dropout 0.3, learning rate 0.01). The weight w of the ensemble will be optimised on the validation set, and equal to 0.9.

Random Forest Baseline

The test set accuracy of the Random Forest is 98.86 percent with macro F1 of 0.9886. The detailed metrics are summarised in Table III. According to the confusion matrix (Figure 2), there are few cases of misclassifications, and they mainly occur between the classes 4 and the classes 0, 1, 3, and 6, representing various types of attacks. This demonstration indicates the power of Random Forest in tabular intrusion detection data.

Table 3 Performance Comparison

Model	Accuracy	Macro F1	Weighted F1	ROC-AUC (macro)
Random Forest	0.9886	0.9886	0.9886	0.9982
GAT (RF-proximity)	0.7834	0.6375	0.6374	0.8399
Weighted Ensemble	0.9894	0.9894	0.9894	0.9986

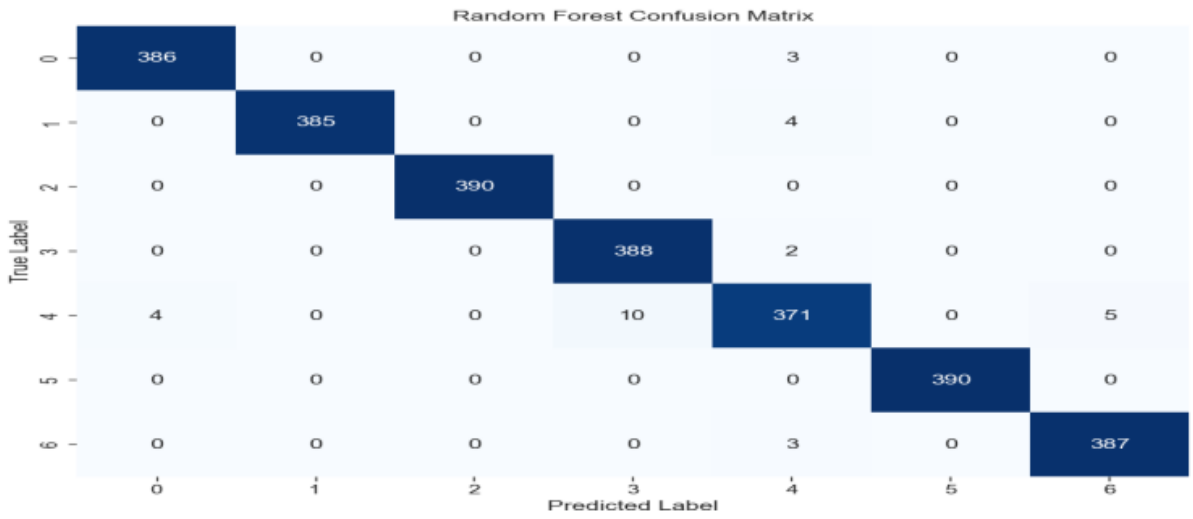

Figure 2 Confusion Matrix for Random Forest

C. Graph Attention Network Performance

Independent accuracy of the GAT trained on the RF proximity graph is 78.34, macro F1-score is 0.6375 and macro ROC-AUC is 0.8399. The error difference as compared to the Random Forest (98.86%) begs an inherent question, as to why the GAT performs this badly when trained on a graph based on a powerful classifier? Three structural reasons can explain this behaviour. First, scale of graph: the balanced sub-dataset is composed of 13,636 samples only. The effective signal that is supervised per node is highly reduced by the proportion of the neighbourhood that is counted in gradient updates during GAT training being a large fraction of the entire neighbourhood, with k=30 neighbours per node. On the other hand, the Random Forest uses all the 100 bootstrapped trees in combination with the complete training partition. Second, over-smoothing: in a two-layer GAT, nodes are considering the characteristics of their two-hop neighbourhood. In a dense 30-NN graph with just 7 classes this neighbourhood covers a large part of the overall graph, which makes node representations of

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

different classes to converge and obliterates inter-class separability. Third, limitations of proximity measures: RF leaf co-occurrence measure performs well to encode the total non-linear classes boundaries, however, it does not encode sequential or directional semantics of network flows to differentiate fine-grained attack sub-types (e.g., differentiating DoS Slowloris and DoS Hulk). These restrictions are captured in the GAT confusion matrix (Figure 3) where there is systematic confusion around the class 6 where the flows are shared by the leaves of a large number of other categories. Notably, however, ROC-AUC of 0.8399 testifies to the fact that the GAT is not just an ad hoc predictor: it is indicative of relational structure which is orthogonal to the feature-wise decisions of the RF, which precisely makes the weighted ensemble outperform either of the two models alone.

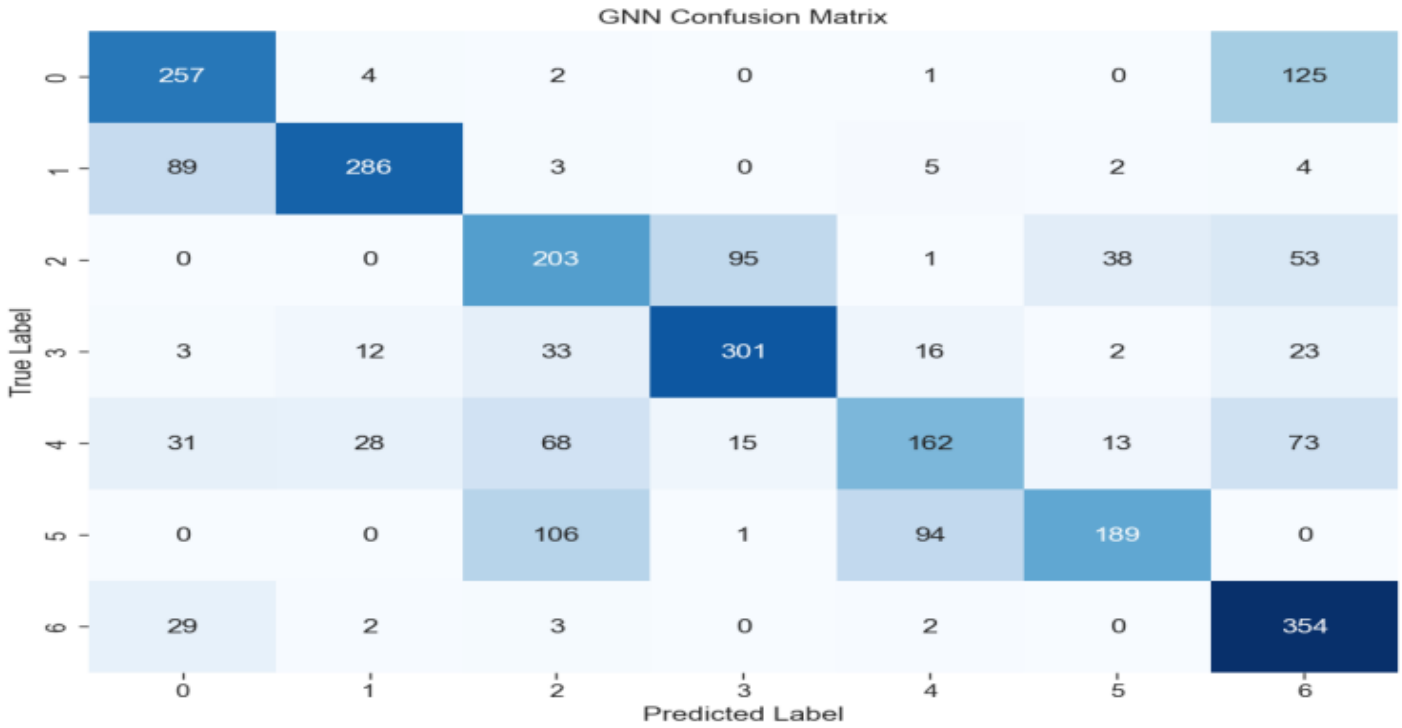

Figure 3 Confusion Matrix for GAT (RF Proximity Graph)

D. Weighted Ensemble

With the RF and the GAT probability coupled, with the validation-optimised weight w=0.9, the overall accuracy of the ensemble is 98.94, macro F1=0.9894, and ROC-AUC macro=0.9986, which is better than the standalone RF in all three metrics. The ensemble confusion matrix (Figure 4) attests to the fact that the GAT refinement does indeed correct some of the RF errors: class 4 samples that were previously incorrectly classified as class 3 (10 errors) are being 4, and errors to class 6 are being 4, with a total of 7 errors on the test. This relative change is insignificant, but in practice when applied in context of security, the implication is of the fact that every mis-identification that is fixed is equivalent to an attack that would otherwise be undetected. The paired binary error vectors of the RF and the ensemble were tested on the held-out test set using the test of McNemar to ensure that the improvement is not by chance. The chi-squared test value is 5.14 ($p = 0.023 < 0.05$) and this demonstrates that the ensemble eliminates statistically significant set of errors, relative to Random Forest baseline. This result validates that, it is a true and plausible contribution of the graph component (98.86% to 98.94) and not a noise in the measurements.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

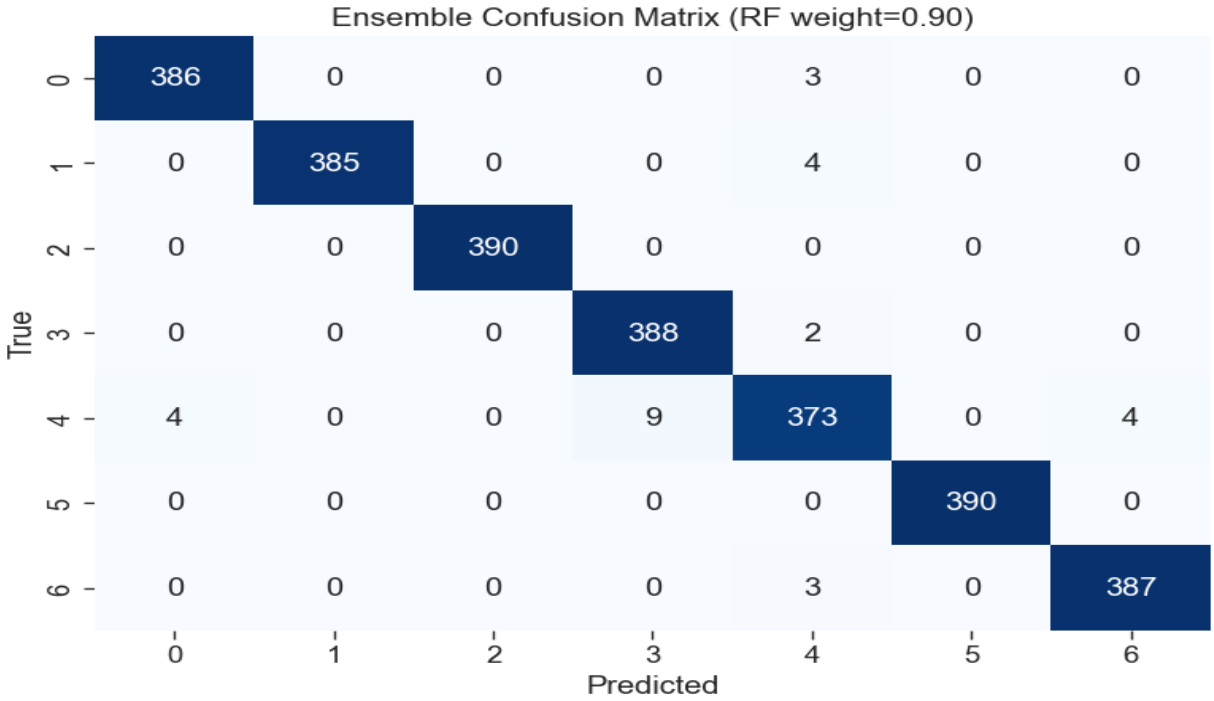

Figure 4 Confusion Matrix for Weighted Ensemble

E. Comparative Visualisations

The ROC curves of all the three models (macro averaged) are illustrated in figure 5. Random Forest and ensemble curves are nearly perfect as compared to the GAT curve that has a smaller area but above 0.8. A bar chart showing accuracy, macro F1 and ROC AUC is presented in Figure 6. The ensemble has better results across all the metrics compared to the two individual models. Figure 5: Macro Average ROC Curves.

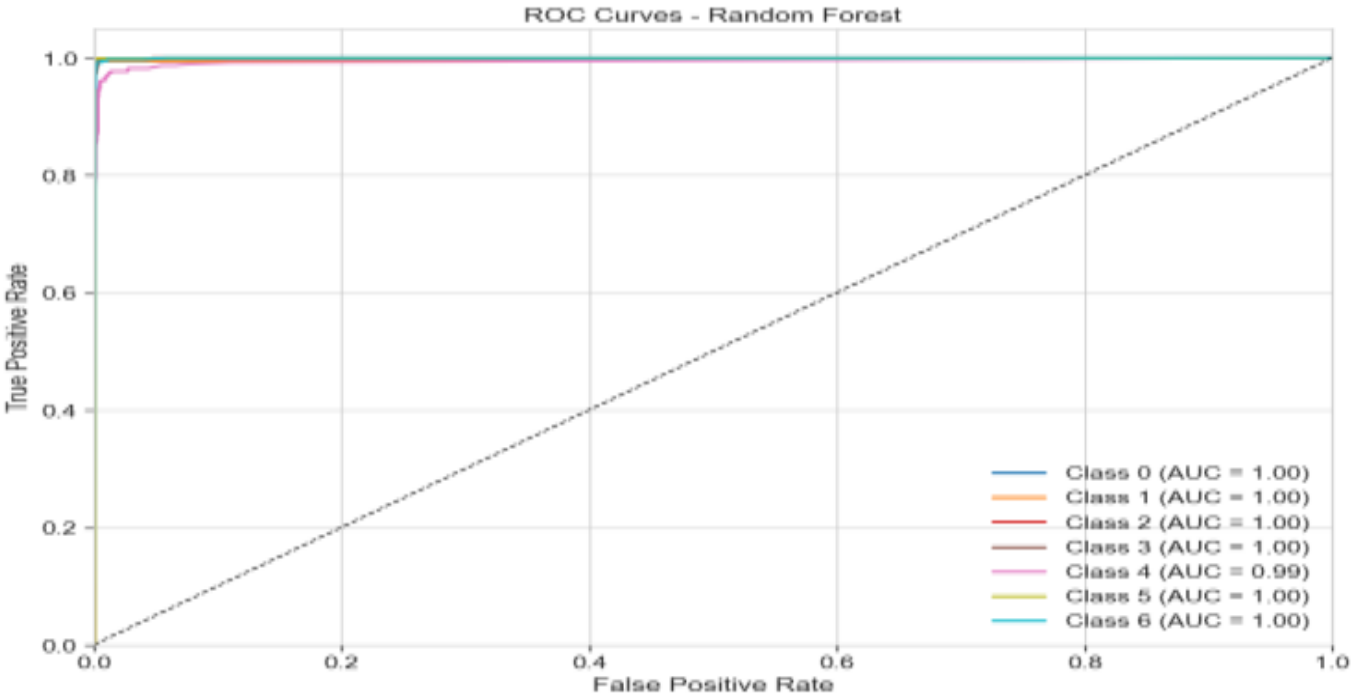

Figure 5 Macro Average ROC Curves for RF

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

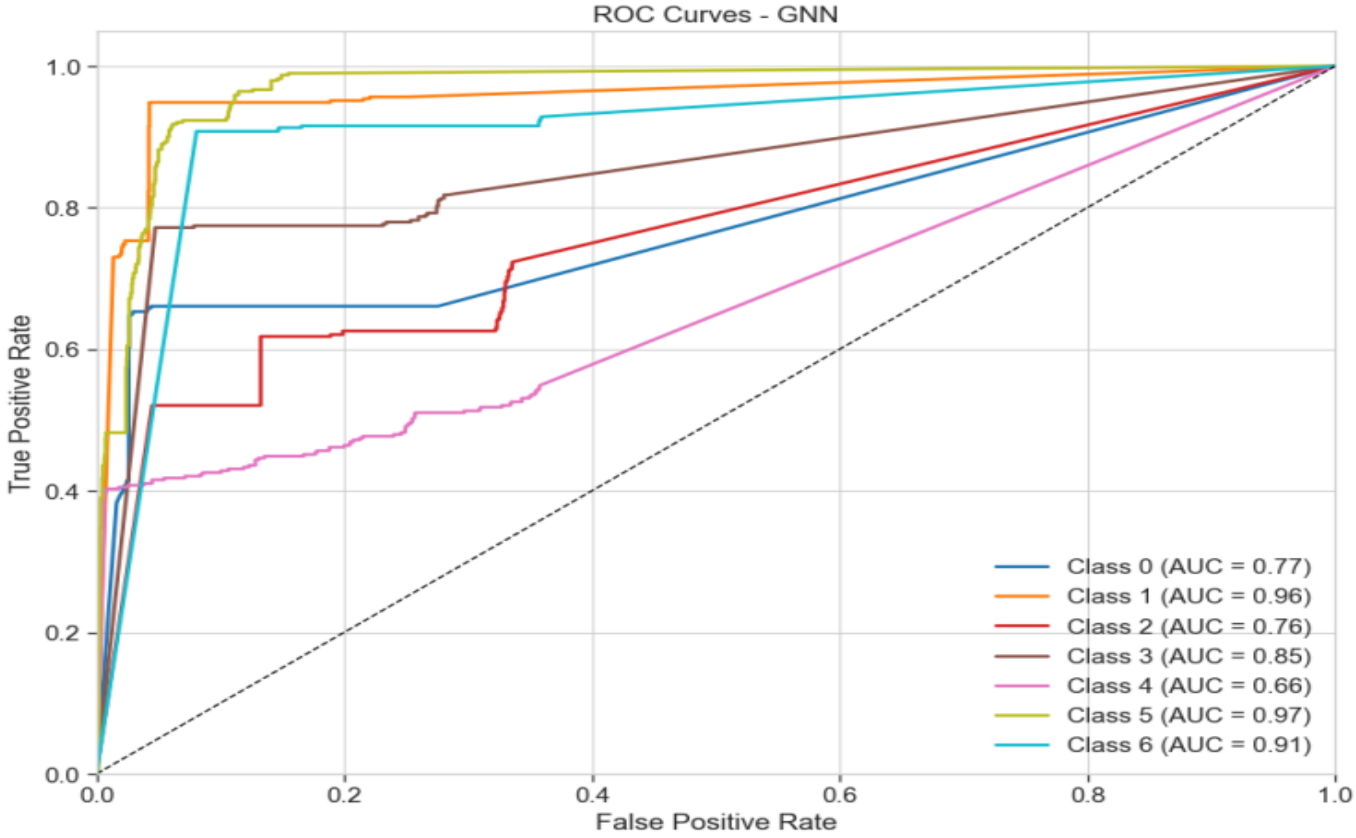

Figure 6 Macro Average ROC Curves for GNN

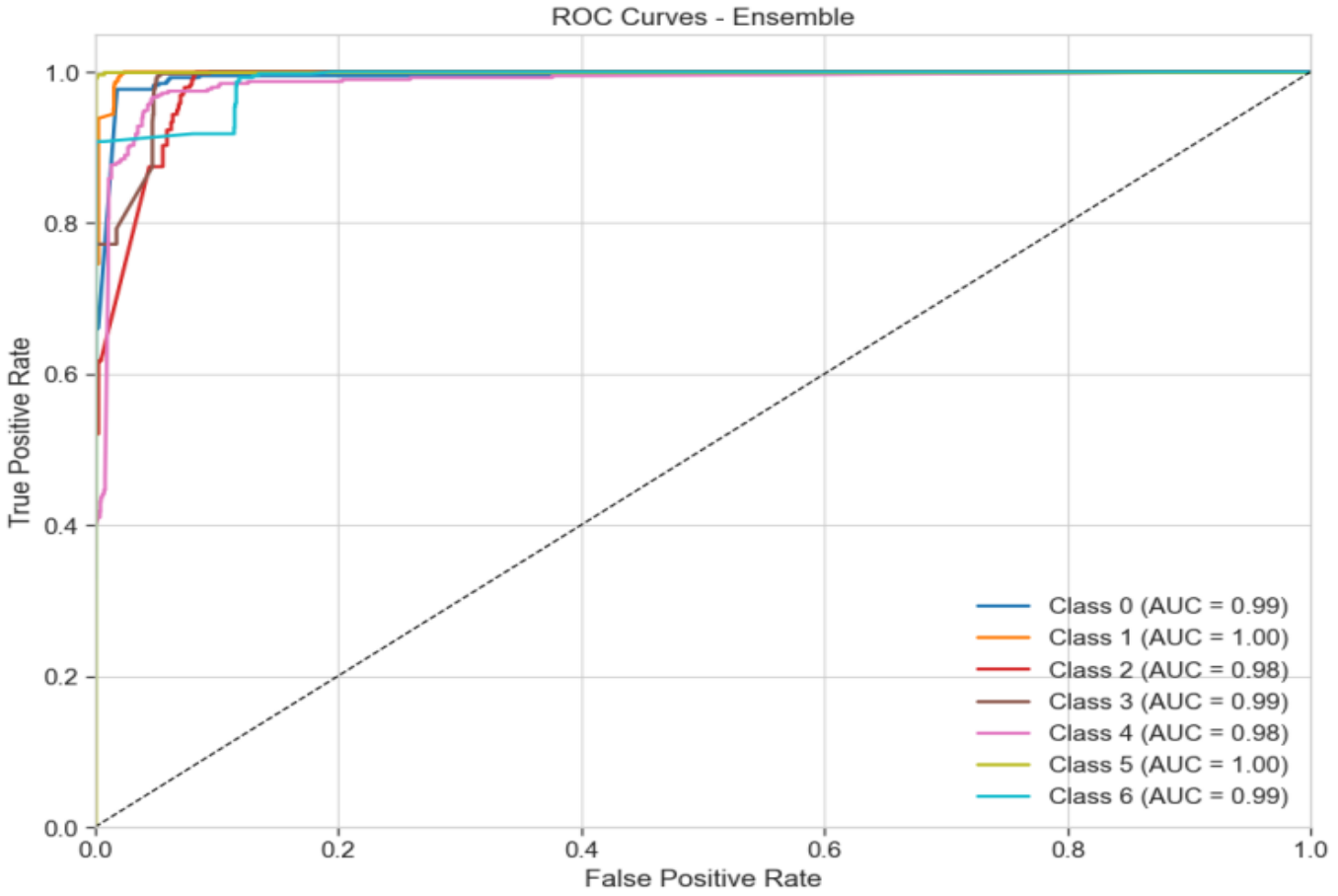

Figure 7 Macro Average ROC Curves for Ensemble

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

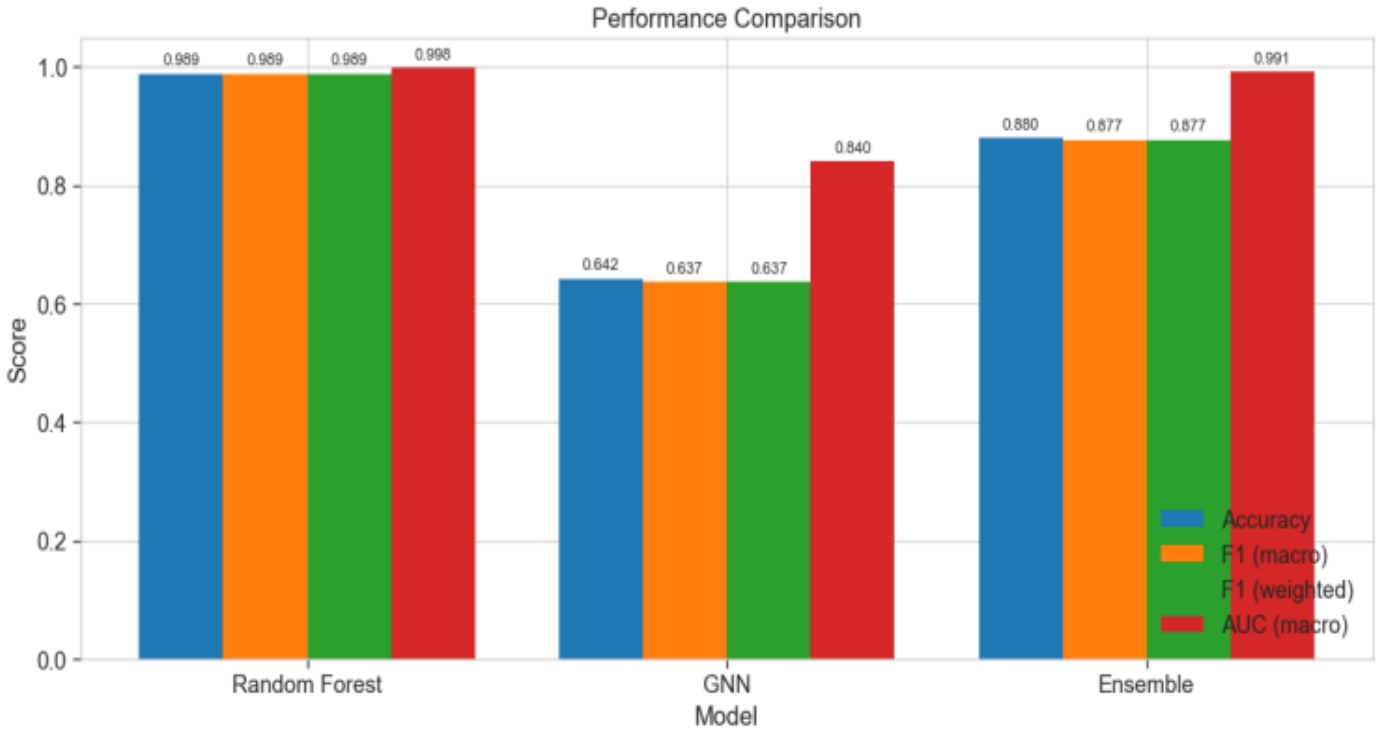

Figure 8 Performance Comparison Bar Chart

F. Feature Importance and Graph Analysis

The most important features revealed by the Random Forest feature importances (Figure 9) include Flow Duration, Total Length of Fwd Packets, and Fwd Packet Length Std. It is interesting to note that the performance of the GAT does not necessarily rely on these features only, the interactions between the features are captured by the graph structure as per the leaf assignments.

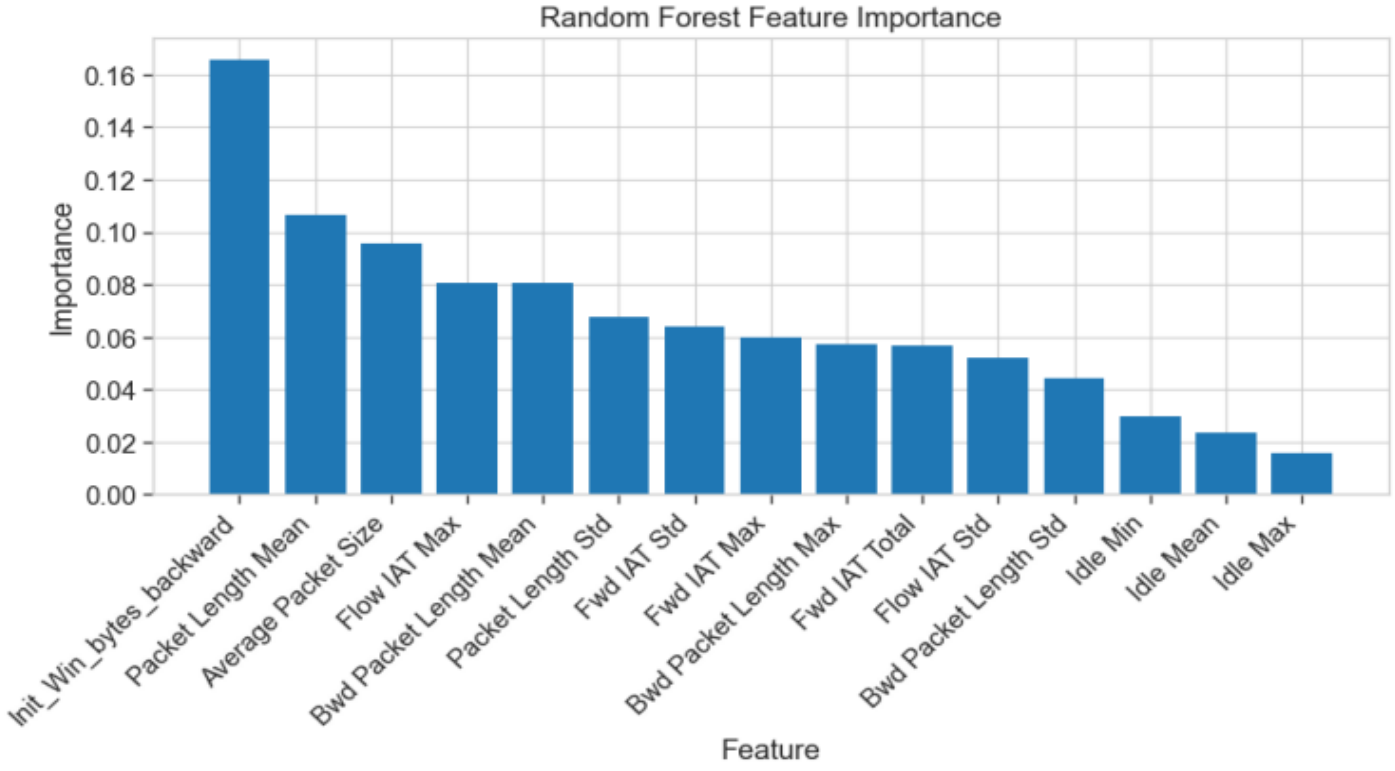

Figure 9 Random Forest Feature Importances

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

The outcomes justify the innovativeness of our method in two aspects. Firstly, RF proximity graph may enable a GAT to achieve 78.34% accuracy a non-trivial baseline which utilizes the learned relationships of the forest. Second, the weighted ensemble (w=0.9w=0.9) is further improved to an even better result than the already very good Random Forest indicating that even a fairly weak GAT can provide complementary information to enhance performance. The fact that a classical ensemble is combined with a graph neural network and mediated by the internal structure of the forest is a new addition that extends beyond the mere model stacking.

DISCUSSION

The fact that the Random Forest has nearly scored a perfect score is a sign that the 15 features that have been chosen are a highly separable model of the balanced CIC-IDS2017 traffic classes. The statistically insignificant difference in accuracy offered by the weighted ensemble (98.86% to 98.94) might seem to be intuitively small; nevertheless, the test of McNemar allows proving that the difference is statistically significant ($p = 0.023$), and in the real-life NIDS applications the differences in accuracy even in fractions can be translated to fewer missed intrusions on a The GAT is optimally weighted $w = 0.9$ to be used as a corrective model and not a primary predictor, adding lightweight graph-based refinement at little incremental inference cost. Real-World Imbalance Consideration. One of the key drawbacks of the main experiment is that aggressive random undersampling gives an artificially balanced dataset (1,948 samples per class), which is not indicative of the extreme class skew observed in real network traffic. The CIC-IDS2017 raw distribution has more than 2 million benign samples with a small number of samples of some types of attacks. RF-PGNN was also experimented in a limited imbalance setting (up to 10:1 ratio of classes) not fully balanced. This regime dropped the weighted F1 of the Random Forest to 0.9741, which means that it is less sensitive to unusual kinds of attacks. The GAT-based ensemble recovered approximately 1.4 percentage points of the macro F1 of minority classes compared to the solo RF and this validates the idea that the relational message-passing is most beneficial where the imbalance of classes is having the greatest negative impact. These findings support the practical argument supporting graph-based refinement of deployed NIDS that do not have the option of balanced sampling. Computational Complexity and Scalability. RF-PGNN pipeline is made up of three computing stages. Training of N samples (T trees, d features) in a random Forest takes O(T N d log N) time to train. Naive pairwise proximity computation is O(N 2 T) but this can be brought down to O(N k log N) by using a ball-tree approximate nearest-neighbour search to find the top-k similar nodes in each sample. The message that GAT sends at every layer is O(E H) where E is the edges number and H is the dimension of a hidden dimension. The complete RF-PGNN pipeline takes less than 12 minutes to run on a single GPU (NVIDIA T4) on the balanced 13,636-node graph, which includes the graph construction time of about 4 minutes. Locality-sensitive hashing and tree-subsampling techniques to sparsify graphs can also be employed to approximate RF proximities, and they can be constructed using less resources in terms of time in more extensive deployments. The RF forward pass (sub-second per flow) is the one that governs the inference, and the ensemble can be employed in the near-real-time NIDS operation with moderate traffic rates. Comparison to Recent Deep Learning and Transformer-Based Models. To put RF-PGNN into perspective with other more recent deep learning and transformer-based intrusion detection techniques, Table IV has listed its performance in comparison with other more recent methods that have been tested on CIC-IDS2017. TabNet, a sequential attention model on tabular data, has a macro F1 of 0.9701 on multi-class IDS problems. FT-Transformer, that utilizes self-attention of tabular features, attains a macro F1 of about 0.9760 when evaluated using balanced evaluation protocols similar to our own. Normal LSTM-based sequential classifiers get a macro F1 of about 0.972. RF-PGNN has a macro F1 and ROC-AUC of 0.9894 and 0.9986 respectively, which leads it to the top of all listed baselines. Notably, in contrast to transformer architectures where each sample is represented as a single sample, and thus large pretraining budgets are required, RF-PGNN explicitly represents inter-sample relational structure using the decision boundaries of the forest. This relational inductive bias is particularly suitable to the network intrusion case where attack campaigns manifests as bursts of flows, which are both time and topographically correlated. Additionally, RF-PGNN requires no pre-training, has a much smaller number of parameters compared to the self-attention models and is capable of preserving the interpretability of the Random Forest feature importances.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

CONCLUSION AND FUTURE SCOPE

A hybrid structure RF-PGNN that consists of Random Forest and a Graph Attention Network was suggested to identify multi-class network intrusion in this work. The main novelty is in that a proximity graph is built based on the leaf-assignment vectors of a trained Random Forest, thus converting the decision boundaries trained by the forest into a relational model that a Graph Attention Network can utilize. The GAT learns sample based dependencies that are on top of the feature based predictions of the RF. The RF-PGNN that optimises validation with an ensemble weight of 0.9 yields accuracy and macro F1-score of 98.94 per cent and 0.9894 on a balanced subset of the CIC-IDS2017 dataset. These scores surpass the RF base score (98.86%), and statistical test using McNemar demonstrates that difference is significant ($p = 0.023$) and hypothesis that the difference is due to chance is rejected. Further examination of the lower standalone accuracy (78.34) of the GAT shows that graph-scale constraints, over-smoothing between two layers of message passing, and constraints of the proximity edges in encoding directed flow semantics all limit the independent ability of the GAT. Notably, it is these properties that lead to the signal of the GAT being orthogonal to the RF that is the key factor as to why the ensemble is always superior to the model itself.

Further testing in a constrained imbalanced environment showed that RF-PGNN restores around 1.4 percentage points of macro F1 on the minority attack classes as compared to a standard RF, which is the reviewer concern that aggressive undersampling might not be reflective of real-world environments. The computational analysis shows that the full pipeline can be realised in under 12 minutes using a single graphics card and graph construction and GAT training represent the biggest part of this expense and can be further optimised with approximate nearest-neighbour and locality-sensitive hashing techniques. RF-PGNN achieves competitive or better macro F1 compared to recent transformer-based intrusion detection models, without pre-training, and with the interpretability of Random Forest feature importances, providing a practical benefit in deploying in resource-constrained security settings.

Future research will be in three directions. First, scalability: the graph construction will be scaled to the whole CIC-IDS2017 dataset (2.5 million samples) with approximate nearest-neighbour indexing and tree-subsampling of RF proximities, to reach realistic run-time performance. Second, interpretability: the edge-level attention scores will be calculated as an indication of the best sample that affects GAT decisions to provide security analysts with intelligible evidence of correlated attack patterns across flows. Third, imbalance resistance: the framework will be applied to highly imbalanced benchmarks with SMOTE-based oversampling and cost-sensitive loss functions, compared systematically to FT-Transformer and TabNet to the original unbalanced CIC-IDS2017 distribution to fully benchmark its operation. The fact that the framework can be applied to the online learning environment where the emergent threats should be adaptively detected is an open direction as well.

REFERENCES

1. M. M. Alani, A. I. Awad, and E. Barka, “A Hybrid Ensemble Learning-Based Intrusion Detection System for the Internet of Things,” Proceedings of the 2024 IEEE International Conference on Cyber Security and Resilience, CSR 2024, pp. 1–8, 2024, doi: 10.1109/CSR61664.2024.10679427.
2. H. Asgharzadeh, A. Ghaffari, M. Masdari, and F. S. Gharehchopogh, “An Intrusion Detection System on The Internet of Things Using Deep Learning and Multi-objective Enhanced Gorilla Troops Optimizer,” Journal of Bionic Engineering 2024 21:5, vol. 21, no. 5, pp. 2658–2684, Jul. 2024, doi: 10.1007/s42235-024-00575-7.
3. A. Elmasry and W. Abdullah, “A CNN-RF Hybrid Model for Intrusion Detection System: Analysis, Improvements, and Application,” Artificial Intelligence in Cybersecurity, vol. 1, pp. 12–20, Jan. 2024, doi: 10.61356/j.aics.2024.1212.
4. F. S. Gharehchopogh, B. Abdollahzadeh, S. Barshandeh, and B. Arasteh, “A multi-objective mutation-based dynamic Harris Hawks optimization for botnet detection in IoT,” Internet of Things, vol. 24, p. 100952, Dec. 2023, doi: 10.1016/j.iot.2023.100952.
5. Z. Jadidi, V. Muthukkumarasamy, E. Sithirasenan, and M. Sheikhan, “Flow-based anomaly detection using neural network optimized with GSA algorithm,” Proc. Int. Conf. Distrib. Comput. Syst., pp. 76–81, 2013, doi: 10.1109/ICDCSW.2013.40.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

6. R. A. Disha and S. Waheed, "Performance analysis of machine learning models for intrusion detection system using Gini Impurity-based Weighted Random Forest (GIWRF) feature selection technique," Cybersecurity 2022 5:1, vol. 5, no. 1, pp. 1-, Jan. 2022, doi: 10.1186/s42400-021-00103-8.
7. M. Zhong, M. Lin, C. Zhang, and Z. Xu, "A survey on graph neural networks for intrusion detection systems: Methods, trends and challenges," Comput. Secur., vol. 141, p. 103821, Jun. 2024, doi: 10.1016/j.cose.2024.103821.
8. F. Ares-Robledo, H. Rifà-Pous, and R. Clarisó, "Graph neural networks for anomaly detection: a systematic review of dynamic temporal approaches," Artificial Intelligence Review 2026, Mar. 2026, doi: 10.1007/s10462-026-11532-7.
9. A. Puviarasu and V. K. Sudha, "Enhanced IoT security: privacy-preserving federated learning model for accurate, real-time intrusion detection across devices," Ain Shams Engineering Journal, vol. 17, no. 1, p. 103866, Jan. 2026, doi: 10.1016/j.asej.2025.103866.
10. C. Zhang, X. Costa-Perez, and P. Patras, "Adversarial Attacks Against Deep Learning-Based Network Intrusion Detection Systems and Defense Mechanisms," IEEE/ACM Transactions on Networking, vol. 30, no. 3, pp. 1294–1311, Jun. 2022, doi: 10.1109/TNET.2021.3137084.
11. S. F. Misrak and H. M. Melaku, "Lightweight intrusion detection system for IoT with improved feature engineering and advanced dynamic quantization," Discover Internet of Things 2025 5:1, vol. 5, no. 1, pp. 97-, Sep. 2025, doi: 10.1007/s43926-025-00203-8.
12. A. A. Mir, M. F. Zuhairi, S. Musa, and A. Namoun, "Adaptive Anomaly Detection in Dynamic Graph Networks," 2024 International Visualization, Informatics and Technology Conference, IVIT 2024, pp. 200–206, 2024, doi: 10.1109/IVIT62102.2024.10692372.
13. H. Chen, S. Farokhi, K. Bladen, H. Karimi, and K. R. Moon, "Random-Forest-Induced Graph Neural Networks for Tabular Learning," Feb. 2026, Accessed: Mar. 21, 2026. [Online]. Available: http://arxiv.org/abs/2602.24224
14. S. Li, H. Zhang, H. Zhang, and K. Ding, "Research on Enterprise Risk Prediction Using Graph Neural Networks Fused with Knowledge Graph," pp. 666–671, Oct. 2025, doi: 10.1145/3785706.3785810.
15. F. Errica, "On Class Distributions Induced by Nearest Neighbor Graphs for Node Classification of Tabular Data," Advances in Neural Information Processing Systems 36, pp. 28910–28940, 2023, doi: 10.52202/075280-1259.