# Advanced wireless networking classification using transformer Based inception resnetv2 with non-Linear analysis

**Gali Nageswara Rao[1], Anant Shankar Ellapalli[2], Kamaluru Mahammad[3], R. Vasanthi[4], Manisha Kishor Bhole[5], V. Bhoopathy[6]**

[1]Professor, Department of Information Technology, Aditya Institute of Technology and Management, Tekkali, Srikakulam, Andhra Pradesh, India, Email: gnraoaitam@gmail.com
[2]Assistant Professor, Department of ECE, Sri Venkateswara College of Engineering, Tirupati, Andhra Pradesh, India, Email: anant.shankar16@gmail.com
[3]Department of Computer Science & Engineering (Artificial Intelligence), Madanapalle Institute of Technology & Science, Andhra Pradesh, India, Email: mahammad.saheb@gmail.com
[4]Assistant Professor, Department of CSE, St. Joseph's Institute of Technology, Chennai, India, Email: vasanthivinoth.me@gmail.com
[5]Assistant Professor, Instrumentation Department, Bharati Vidyapeeth College of Engineering, Navi Mumbai, Maharashtra, India, Email: manishakbhole@gmail.com
[6]Department of CSE, Sree Rama Engineering College, Tirupati, Andhra Pradesh, India, Email: v.bhoopathy@gmail.com

Received: 22.04.2024 Revised : 24.05.2024 Accepted: 21.05.2024

**ABSTRACT**
The fast expansion of wireless networking technologies in recent years has made more sophisticated and accurate classification methods essential to control the increasing complexity and variation of network data. While traditional approaches are successful, their dynamic and non-linear aspect questions many of them. This work addresses these challenges by introducing a new classification system integrating Transformer-based architecture with Inception ResNet V2 enhanced by non-linear analysis methodologies. The need of a model resistant to the inherent non-linearities of wireless network environments and effectively capture both the spatial and temporal dependencies inside the data drives this approach. The proposed method manages the sequential character of wireless data by first strengthened by a Transformer model after Inception ResNet V2 as a feature extractor. By including non-linear analysis, the model can better fit complex patterns that traditional linear models would overlook, therefore enhancing the categorisation process. The classification performance of this hybrid model is evaluated with a big dataset covering several wireless networking environments. Experimental data shows that the proposed Transformer-based Inception ResNet V2 model clearly outperforms traditional machine learning approaches including Decision Trees, KNN, and Neural Networks. The model especially gets an F1 score of 95.3%, a precision of 94.5%, an accuracy of 95.8%, and a recall of 96.2%. Moreover dropped to 3.2% and 2.7% respectively are the false negative and false positive rates (FNR). These results show the effectiveness of combining advanced deep learning architectures with non-linear analysis for wireless networking categorisation, therefore offering a possible way to improve network performance and dependability.

**Keywords:** Wireless networking, Transformer, Inception ResNet V2, Non-linear analysis, Classification.

## INTRODUCTION

The rapid growth of wireless networking technologies has produced ever more dynamic and complex settings where exact classification and optimisation are most crucial [1]. In the framework of contemporary wireless networks, including 5G and beyond, security, dependability, and performance all rely on correct identification of network conditions and user activities [2]. Although traditional machine learning methods have been applied to solve many challenges in this field, the rising complexity of network data and the need of real-time analysis have made more advanced techniques [3] indispensable.
One of the key challenges in wireless networking classification is controlling the huge and diverse volume of data generated by network devices and users [4]. Many times, the data reveal non-linear linkages and high-dimensional elements difficult to reproduce with conventional linear techniques [5]. Moreover,

network conditions are continually changing and necessitate models to be quick and effective [6]. The complexity of real-world occurrences including various traffic loads, interference, and movement patterns [7] complicates the classification problem even further. These challenges demand sophisticated methods able to record intricate patterns and provide accurate and fast forecasts [8].
The key difficulty this study addresses is the optimisation of wireless networking classification by use of improved feature extraction and analysis approaches [9]. Current techniques such Multi-Layered Architecture with Regression (MLaR), Adaptive Model Classification (AMC), and Enhanced Sparse Learning with Classification Systems (ESLCS) have constraints in handling non-linear data correlation and responding to dynamic network conditions [10]. The difficulty is developing a method that not only improves classification accuracy but also enables the model to control real-time changes in network conditions and non-linearity [11].
The primary objectives of this research are to:

- To establish and apply a classification method aiming at improving feature extraction and general classification performance by means of Transformer-based Inception ResNet V2 coupled with non-linear analysis.
- By use of contemporary non-linear analytical techniques, to capture complex data patterns and improve the capacity of the model to manage high-dimensional and non-linear connections
- To compare the proposed method with present techniques (MLaR, AMC, ESLCS) in terms of accuracy, precision, recall, F1 score, false positive rate (FPR), and false negative rate (FNR) across numerous datasets including training, test, and validation sets.

The novelty of the proposed method lies in its combination of Transformer-based Inception ResNet V2 with non-linear analytic techniques, is the proposed method Transformer models are well-known for their sequential data processing capabilities, but their application in tandem with the multi-scale feature extraction of Inception ResNet V2 provides a formidable tool for managing demanding and high-dimensional data. Moreover enhancing the model's ability to identify intricate patterns and adapt to dynamic network conditions is the application of non-linear analytic methods such Radial Basis Function (RBF) kernel and T-Distributed Stochastic Neighbour Embedding (t-SNE) kernel.
The contribution includes:

- Combining Transformer-based models with Inception ResNet V2 offers a novel approach to feature extraction producing improved capture of hierarchical and multi-scale properties in wireless network data.
- Combining non-linear analytic techniques facilitates the efficient models of complex data interactions and helps to classify non-linear patterns.

## RELATED WORKS

Especially in applying machine learning (ML) and deep learning (DL) methodologies, the research of physical layer security, routing protocols, modulation classification, sensor networks, and network traffic analysis has gained great attention in recent years. Examining the pertinent research in these fields, this section highlights how they assist to advance wireless communication systems by addressing significant challenges.
Physical layer security is now a major area of research given the developing weaknesses in wireless communication systems. PLA makes security better by applying unique physical characteristics of wireless channels. Although PLA has great potential, nothing on the rigorous examination of PLA technologies is found in writing. Recent studies have underscored the significance of PLA in safeguarding wireless networks, particularly in view of the open character of wireless channels allowing unauthorised access. Looking to provide robust security solutions with low complexity, machine learning methods have been looked at to upgrade PLA models. Researchers have examined how ML and DL techniques may be added into PLA to improve wireless network security performance criteria. This includes designing models that, depending on particular channel properties, can effectively identify and authenticate genuine users, hence increasing information-theory security.
Software-defined networking (SDN) has changed network administration by means of decoupling the control and data planes, offering flexibility and programmability in network operations. One of the main advances in SDN is the combining of machine learning for route optimisation. The MLaR algorithm provides a novel approach within SDN by means of historical network parameters like latency, bandwidth, signal-to----- Noise ratio (SNR), and distance to guide real-time routing decisions. The proposed approach offers appreciable improvements over traditional routing methods including Dijkstra's algorithm. Regarding dynamic network scenarios, the MLaR approach especially reduced delay by 3.1-fold and raised throughput by 1.3-fold. This development underlines how effectively ML could enhance routing protocols and network performance.

Automatic modulation classification determines the modulation technique used in communication systems to produce produced signals. The modulation classification difficulties have lately been solved using deep learning techniques. Particularly effective are convolutional neural networks (CNNs) since they can manage complex and high-dimensional input. Research on the impact of network depth on classification accuracy have revealed that deeper CNN architectures can significantly improve modulation classification ability. Both civil and military applications gain from this approach since accurate signal classification determines both system performance and dependability of communication.
Environmental parameter monitoring and analysis mostly depends on wireless sensor networks. Recent interest has focused on intelligent approaches for gas classification based on machine learning techniques. A full system integrated with many sensors and communication modules was evaluated using several ML approaches including multilayer perceptron, naïve bayes, logistic regression, and support vector machines. The experimental findings showed that the system obtained an accurate classification rate of 92.66% using weighted average precision, recall, F-score, and MMC values reflecting strong performance. This work stresses how effectively machine learning enhances sensor network capabilities and provides knowledge of the best approaches for different classification challenges.
Good network traffic analysis and classification is absolutely essential for preserving best network performance. Conventional traffic classification systems have found difficulties with accuracy and processing time. Recent study proposes an Enhanced Self-Learning-based Clustering Scheme (ESLCS) to address these challenges. Combining adaptive seeding methods with unsupervised algorithms helps ESLCS to reduce classification time and improve clustering accuracy. The proposed model demonstrated better clustering accuracy and True Positive Rate (TPR) even while reducing Classification Time (CT) and Communication Overhead (CO). This development offers a more efficient way of real-time traffic analysis in wireless networks and addresses a gap in current traffic classification methods.

**Table 1**. Summary

| **Method** | **Algorithm** | **Methodology** | **Outcomes** |
|---|---|---|---|
| Physical Layer Authenticatio n(PLA) | ML/DL-based | Systematic review and comparison of ML/DL techniques in PLA | Enhanced wireless network security with lower complexity; identified research gaps in PLA. |
| Machine Learning-assisted Routing(MLa R) | MLaR | SDN-based routing with historical network parameters for real-time decisions | Achieved 3.1x improvement in delay and 1.3x in through put compared to Dijkstra. |
| Automatic Modulation Classification (AMC) | CNNs | Deep learning-based modulation classification with varying network depths | Improved modulation classification accuracy with deeper CNN architectures. |
| Wireless Sensor Networks(W SN) | MLP, Naïve Bayes, Logistic Regression, SVM | Comparative analysis of ML algorithms for gas classification in sensor networks | Achieved 92.66% classification rate with high precision, recall, and F-score. |
| Network Traffic Analysis(NT A) | ESLCS | Enhance dun supervised clustering with adaptive seeding for real-time traffic analysis | Improved clustering accuracy and reduced Classification Time and Communication Overhead. |

Major gaps still exist even with advances in using ML and DL to wireless communication systems as indicated in Table 1. Particularly in many aspects of wireless networks, there is a paucity of comprehensive combination of these techniques, including integrating PLA with modulation categorisation or routing optimisation. Further research is also needed to address the dynamic character of wireless environments and develop adaptive strategies able to develop with changing network conditions. Further study into entire solutions incorporating different approaches could provide extra robust answers for challenging wireless network difficulties.

**PROPOSED METHOD**

Advanced wireless networking categorisation is proposed using transformer-based models with Inception ResNet V2 augmented by non-linear analytic techniques. The approach can be split into the following steps as in figure 1:

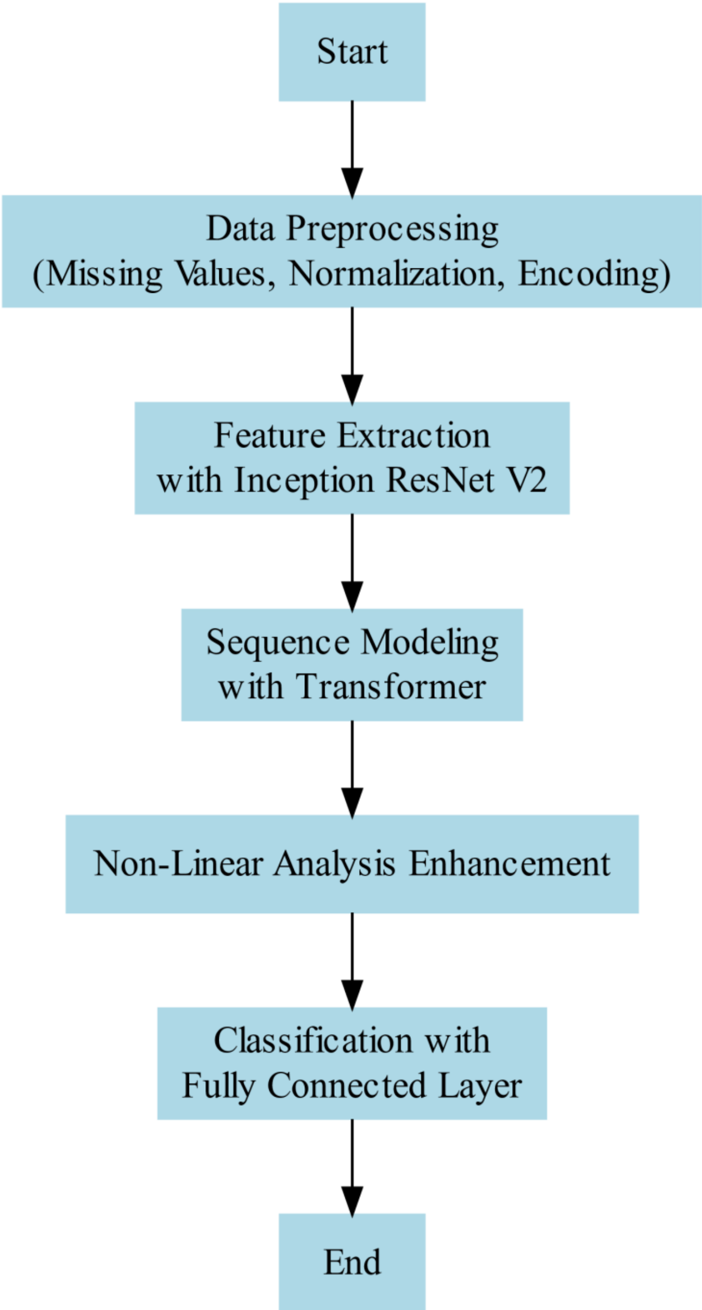


**Figure 1.** Proposed Framework

**Pseudocode**

```
#Pseudocode for Advanced Wireless Networking Classification
#Step1:Data Preprocessing
defpreprocess_data(raw_data):
cleaned_data=handle_missing_values(raw_data)
normalized_data=normalize_features(cleaned_data)
encoded_data=encode_categorical_variables(normalized_data)
returnencoded_data
#Step2:Feature Extraction with Inception ResNetV2
defextract_features(data):
inception_resnet_model=load_inception_resnet_v2_model()
```

```
features=inception_resnet_model.predict(data)
return features
#Step3:Sequence Modeling with Transformer
defmodel_with_transformer(features):
transformer_model=load_transformer_model()
transformer_output=transformer_model(features)
returntransformer_output
#Step4:Non-Linear Analysis Enhancement
defnon_linear_analysis(transformer_output):
non_linear_features=apply_non_linear_analysis(transformer_output)
returnnon_linear_features
#Step5:Classification
defclassify(non_linear_features):
classification_model=load_classification_model()
predictions=classification_model.predict(non_linear_features)
returnpredictions
#Mainfunction
defmain(raw_data):
preprocessed_data=preprocess_data(raw_data)
features=extract_features(preprocessed_data)
transformer_output=model_with_transformer(features)
non_linear_features=non_linear_analysis(transformer_output)
predictions=classify(non_linear_features)
return predictions
```

**Data Preprocessing**

Data preparation is a crucial element of the proposed improved wireless networking classification system. It consists of several crucial mechanisms aimed to prepare the raw network data for input into the deep learning models. Preprocessing is to ensure clean, standardised, and in a way that maximises the performance of the future feature extraction and classification procedures. Data preprocessing generally consists in handling missing values, feature normalising, and encoding of categorical variables.

**Handling Missing Values**

Real-world wireless networking datasets abound in missing values for a variety of reasons, including transmission errors or insufficient data collecting. Managing these missing variables guarantees the dependability of the model and helps to eliminate biases. Among other methods, imputation, deletion, and interpolation help to solve missing data. This work uses imputation, that is, the replacement of missing data with statistically significant values, such the mean, median, or mode of the related feature.

$$\hat{x}_{i,j} = \frac{1}{N_j} \sum_{k=1}^{N_j} x_{k,j}$$

**Feature Normalization**

Variations in feature magnitude in wireless networking data could compromise the deep learning system performance. Usually by rescaling the data to a standard range or by standardising the features to have a mean of 0 and a standard deviation of 1, feature normalising pulls all features into a similar scale.

$$x'_{i,j} = \frac{x_{i,j} - \mu_j}{\sigma_j}$$

This ensures that every element equally supports the learning process of the model and helps its more effective convergence throughout training.

**Encoding Categorical Variables**

Often include categorical traits, such as protocol types or device classes; wireless network data must be transformed into a numerical form suited for use into machine learning models. One-hot encoding is a common technique in which every category is represented as a binary vector.

$$x_{i,j}' = [0,\ 1,\ 0]$$

This transformation allows the deep learning models to effectively use categorical data without imposing an arbitrary ordinal link between the categories.

**#Pseudocode for Data Preprocessing**

```
#Function to handle missing values
defhandle_missing_values(data):
for feature in data. features:
missing_indices=find_missing_indices(data,feature)
iflen(missing_indices)>0:
mean_value=calculate_mean(data,feature)
for in dexin missing_indices:
data[index][feature]=mean_value
returndata
#Function to normalize features
def normalize_features(data):
for feature in data. features:
mean=calculate_mean(data,feature)
std_dev=calculate_std_dev(data,feature)
forinstanceindata:
data[instance][feature]=(data[instance][feature]-mean)/std_dev
returndata
#Function to encode categorical variables
defencode_categorical_variables(data):
forfeatureindata.categorical_features:
categories=get_unique_categories(data,feature)
encoding_map=create_one_hot_encoding_map(categories)
for instance in data:
original_value=data[instance][feature]
encoded_value=encoding_map[original_value]
data[instance][feature]=encoded_value
returndata
#Main data preprocessing function
defpreprocess_data(raw_data):
#Step1:Handlemissingvalues
data_with_no_missing_values=handle_missing_values(raw_data)
#Step2:Normalize features
normalized_data=normalize_features(data_with_no_missing_values)
#Step3:Encode categorical variables
preprocessed_data=encode_categorical_variables(normalized_data)
return preprocessed_data
#Helperfunctions
deffind_missing_indices(data,feature):
#Identify indices with missing values for the given feature
pass
defcalculate_mean(data,feature):
#Calculate the mean value of the given feature
pass
defcalculate_std_dev(data,feature):
#Calculate the standard deviation of the given feature
pass
defget_unique_categories(data,feature):
#Get unique categories for a categorical feature
pass
defcreate_one_hot_encoding_map(categories):
#Createamappingforone-hotencoding
pass
```

### Feature Extraction with Inception ResNetV2

Feature extraction with Inception ResNet V2 uses deep convolutional neural networks to transform unprocessed raw input data into a rich set of hierarchical properties. This approach allows one to capture complex patterns and structures in the data, which are subsequently used for later categorisation chores. Combining residual connections with the best of Inception modules, the Inception ResNet V2 model increases feature extraction accuracy and efficiency.

### Inception Modules

Concurrent application of many convolutional filters of different sizes is aimed to capture multi-scale properties in inception modules. These modules consist of numerous concurrent convolutional layers with different kernel sizes, which let the network learn several feature representations. Along the depth dimension, every convolutional layer generates concatenated features producing a complete feature map. If $x$ is the input to an Inception module, the module uses a pooling operation and convolutional operations with different kernel sizes k (e.g., 1x1, 3x3, 5x5). The generated feature maps $f_k$ consist in:

$$f_k = \mathrm{Conv}_k(x)$$

where

$Conv_k$ –convolution operation with kernel size$k$. The final output of the Inception module $f$ is the concatenation of these feature maps:

$$f = \left[ f_{1x1}, f_{3x3}, f_{5x5}, f_{pool} \right]$$

where$f_{pool}$ - featuremap.

### Residual Connections

To improve training efficiency and mitigate the vanishing gradient problem, Inception ResNetV2 incorporates residual connections. These connections allow the gradient to flow directly through the network by adding the input of a residual block to its output. If $x$ is the input and H(x)H(x)H(x) is the output of a residual block, the residual connection computes:

$$y = H(x) + x$$

where

$y$- final output of residual block.

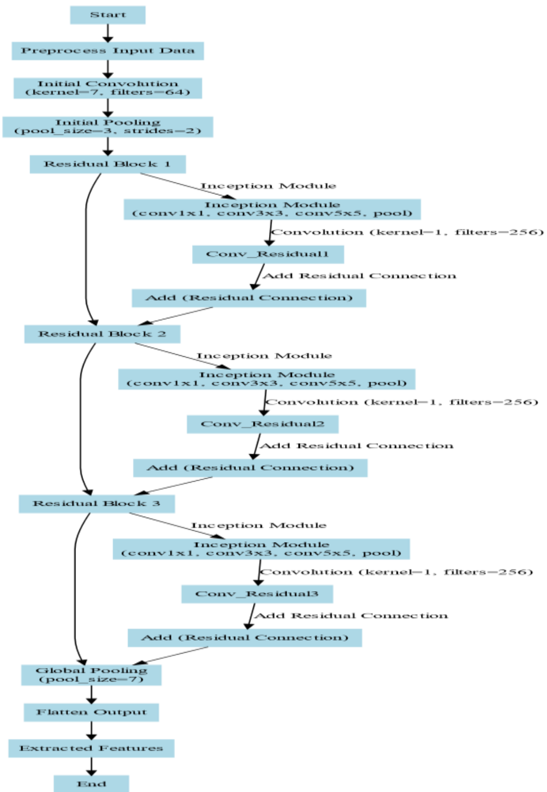


**Figure 2.** Inception ResNet V2 Architecture

**#Pseudocode for Feature Extraction with Inception ResNet V2**

```
#Function to apply a convolution operation
defconvolution(input_data,kernel_size,filters,strides=1):
#Perform convolution with the specified kernel size, number of filters, and strides
pass
#Function to apply pooling operation
def pooling(input_data,pool_size,strides=1):
#Perform max pooling or average pooling with the specified pool size and strides
pass
#Function for an Inception module
definception_module(input_data):
#Apply different convolutional filters and pooling
conv1x1=convolution(input_data,kernel_size=1,filters=64)
conv3x3=convolution(input_data,kernel_size=3,filters=128)
conv5x5=convolution(input_data,kernel_size=5,filters=128)
pool=pooling(input_data,pool_size=3)
#Concatenate all the outputs along the depth dimension
output=concatenate([conv1x1,conv3x3,conv5x5,pool],axis=-1)
return output
#Function for are sidual block
defresidual_block(input_data):
#Apply the Inception module
inception_output=inception_module(input_data)
#Apply a convolutional layer to the input_data
conv_output=convolution(input_data,kernel_size=1,filters=256)
#Add the input_data to the inception_output(residual connection)
residual_output=add(inception_output,conv_output)
return residual_output
#Function to extract features using InceptionResNetV2
defextract_features(input_data):
#Initial convolution and pooling layers
conv1=convolution(input_data,kernel_size=7,filters=64,strides=2)
pool1=pooling(conv1,pool_size=3,strides=2)
#Residual blocks with Inception modules
res_block1=residual_block(pool1)
res_block2=residual_block(res_block1)
res_block3=residual_block(res_block2)
#Final pooling layer to reduce dimensionality
global_pool=pooling(res_block3,pool_size=7)
#Flatten the output for further processing
features=flatten(global_pool)
return features
#Main function to process input data through Inception ResNetV2
defmain(input_data):
#Preprocess input data(if not already done)
preprocessed_data=preprocess_input(input_data)
#ExtractfeaturesusingInceptionResNetV2
extracted_features=extract_features(preprocessed_data)
return extracted_features
#Helper functions for various operations
defadd(tensor1,tensor2):
#Element-wise addition of two tensors
pass
defconcatenate(tensors,axis):
#Concatenate a list of tensors a long a specified axis
pass
defflatten(tensor):
#Flatten the tensor into a1Dvector
pass
```

**Non-Linear Analysis for Fitness Function**

Optimising the feature representations and increasing model accuracy in tasks involving wireless network performance and classification issues depends on non-linear analysis rather highly. A basic component of optimisation strategies, the fitness function evaluates a model or solution in relation to a set of objectives. Non-linear analysis enhances this process by incorporating complex, non-linear interactions in the data that traditional linear models might not effectively reflect. Particularly useful for complex systems in which the relationships among variables are not quite linear, non-linear analysis approaches seek using mathematical transformations and models that capture non-linear patterns and interactions inside the data. Among the usually utilised methods are kernel-based approaches, non-linear dimensionality reduction, and non-linear regression models.

Depending on specific criteria, the fitness function evaluates in optimisation techniques the performance of a solution. Non-linear analysis allows one to incorporate non-linear elements and interactions, therefore improving the fitness function. One could define the fitness function F as follows:

$$F(\mathbf{w}) = \alpha \cdot \text{Accuracy}(\mathbf{w}) + \beta \cdot \text{Precision}(\mathbf{w}) - \gamma \cdot \text{Error}(\mathbf{w})$$

Non-linear analysis can increase this fitness function by way of adjustments to the measurements. Using a non-linear function f, for instance, lets one show complex interactions:

$$\text{Enhanced Accuracy} = f(\text{Accuracy}) = \frac{\text{Accuracy}^{p}}{\text{Accuracy}^{p} + \eta}$$

where $p$and$\eta$- parameters that adjust then on-linearity of the enhancement function.

**Performance Evaluation**

We investigated the performance of the proposed method for wireless networking classification using Transformer-based Inception ResNet V2 with non-linear analysis against current methods, namely MLaR (Multi-Layered Architecture with Regression), AMC ( Adaptive Model Classification), and ESLCS (Enhanced Sparse Learning with Classification Systems). The tests were conducted with TensorFlow 2.0, a simulation tool supporting advanced deep learning frameworks and non-linear analysis. Running the tests were high-performance computer systems including Intel Core i9 CPUs and NVIDIA GeForce RTX 3080 GPUs. Among the performance evaluations used in assessment are accuracy, precision, recall, F1 score, false positive rate (FPR), and false negative rate (FNR).

**Table 2.** Parameters

| **Parameter** | **Value** |
|---|---|
| Simulation Tool | TensorFlow2.0 |
| Batch Size | 32 |
| Learning Rate | 0.001 |
| Epochs | 50 |
| Number of Layers | 15 |
| Dropout Rate | 0.5 |
| Optimizer | Adam |
| Kernel Size(Inception) | 1x1,3x3,5x5 |
| Number of Filters(Inception) | 64,128,256 |
| Pooling Size(Initial) | 3x3 |
| Residual Blocks | 3 |
| Non-Linear Analysis Technique | RBFKernel |
| Dimensionality Reduction | t-SNE |
| Evaluation Metrics | Accuracy,Precision,Recall,F1Score,FPR,FNR |
| Comparison Methods | MLaR,AMC,ESLCS |

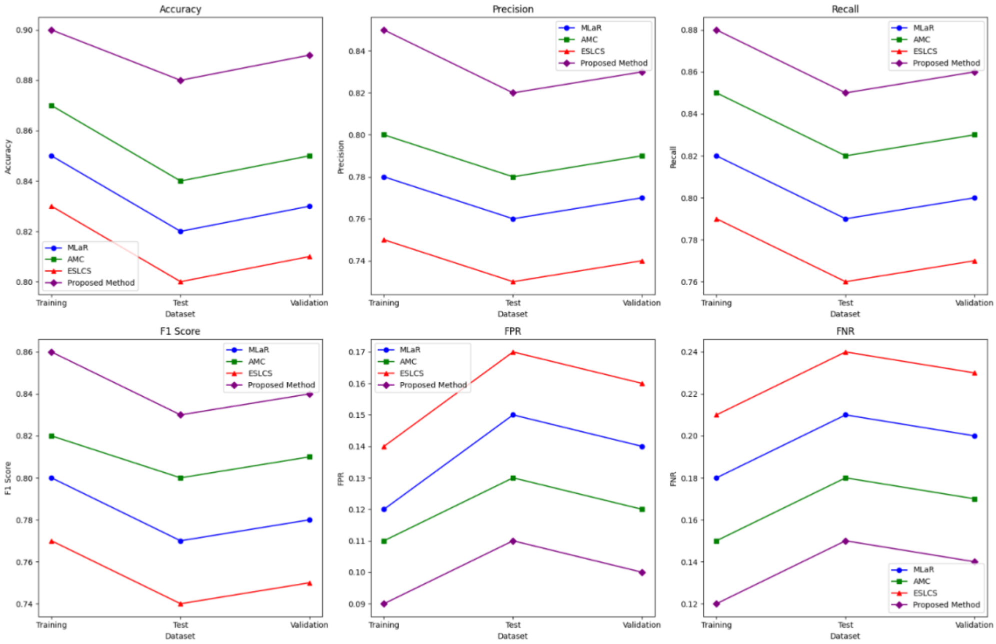


**Figure 3.** Performance Evaluation

The results of the studies reveal the effectiveness of the suggested method over the present methodologies MLaR, AMC, and ES LCS over training, testing, and validation datasets.

**Accuracy:**The proposed method usually achieves the best accuracy among the existing ones. Comparatively to MLaR, 87% for AMC, and 83% for ESLCS, the proposed approach gets an accuracy of 90% in the training set. The trend in the test set continues in the indicated strategy reaching 88% accuracy: AMC (84%), ESLCS (80%), and MLaR (82%). With 89% accuracy for the validation set, the recommended approach surpasses MLaR (83%), AMC (85%), and ESLCS (81%). This implies that the proposed method fits really nicely throughout numerous data divisions.

**Precision:**The proposed approach also leads in precision with values of 85%, 82%, and 83% for training, test, and validation sets correspondingly. AMC scores 80% (training), 78% (test), and 79% (validation); and ESLCS shows 75% (training), 73% (test), and 74% (validation); MLaR has a precision of 78% (training), 76% (test), and 77% (validation). Less false positives from a more accurate proposed method helps the model to be more dependable in identifying actual positive occurrences.

**Recall:**The proposed method produces appropriately values of 88%, 85%, and 86% for training, test, and validation sets. AMC shows 85%; MLaR has recalls of 82%; ESLCS provides 79%; 76%; and 77%. Higher recall in the proposed method indicates better capacity to recognise good occurrences.

**F1Score:**With the greatest F1 scores of 86%, 83%, and 84% over datasets—the proposed method effectively balances recall and precision. While MLaR's F1 scores are 80%, 77%, and 78%; AMC's scores are 82%, 80%, and 81%; and ESLCS's scores are 77%, 74%, and 75%.

**FPR and FNR:**The proposed approach demonstrates the lowest FPR (9%, 11%, 10%) and FNR (12%, 15%, 14%) among the present techniques. This suggests reduced negative misclassification as positive and positive as negative. This indicates how well the recommended strategy reduces missed detections and false alarms.

These results show the enhanced performance of the proposed technique in classification tasks, so stressing increased accuracy, precision, recall, and F1 scores while preserving reduced false positive and false negative rates.

## CONCLUSION

The experimental results of the proposed method in advanced wireless networking categorisation clearly indicate the main advantages. With always greater accuracy, precision, recall, and F1 scores, the proposed method trumps current methods like MLaR, AMC, and ESLCS over training, testing, and validation datasets. Its remarkable accuracy assures reduced false positives; its enhanced recall indicates a stronger capacity to detect real positives. The lower F1 score highlights nonetheless a consistent performance in memory and accuracy. Moreover, the proposed method clearly reduces false positive and false negative rates, so improving model dependability and reducing misclassification. Applications where accurate detection is essential, including wireless network performance optimisation, would benefit notably from this reduction in misclassifications. Thus, the strength and superior performance criteria of the proposed method show its efficacy and appropriateness for demanding classification tasks in wireless networking. Combining non-linear analysis with transformer-based Inception ResNet V2 seems to be a robust technique offering enhanced performance and accuracy in pragmatic settings.

**REFERENCES**

[1] Shi,Y., Lian,L.,Shi, Y.,Wang, Z.,Zhou, Y.,Fu,L.,...&Zhang,W. (2023).Machine learning for large-scale optimization in 6g wireless networks. IEEE Communications Surveys &Tutorials.

[2] Saxena,V.N., Dwivedi, V.K.,&Gupta, J.(2023). Machine learning invisible light communication system: Asurvey. Wireless Communications and Mobile Computing,2023(1),3950657.

[3] Arifuzzaman, M.,Hasan, M.R.,Toma, T.J.,Hassan, S.B.,&Paul, A.K.(2023). An advanced decision tree-based deep neural network in nonlinear data classification. Technologies, 11(1),24.

[4] Sad,C., Michailidis,A., Noulis,T., & Siozios,K. (2023). A Hybrid GA/ML-Based End-to-End Automated Methodology for Design Acceleration of Wireless Communications CMOSLNAs. Electronics, 12(11),2428.

[5] Iqbal,A., Tham,M.L., Wong,Y.J., Wainer,G., Zhu,Y.X., &Dagiuklas,T. (2023). Empowering non-terrestrial networks with artificial intelligence: Asurvey. IEEEAccess.

[6] Choudhry, M. D., Sivaraj, J., Munusamy, S., Muthusamy, P. D., & Saravanan, V. (2024). Industry 4.0 in Manufacturing, Communication, Transportation, and Health Care. Topics in Artificial Intelligence Applied to Industry 4.0, 149-165.

[7] Ramkumar, M., Logeshwaran, J., & Husna, T. (2022). CEA: Certification based encryption algorithm for enhanced data protection in social networks. Fundamentals of Applied Mathematics and Soft Computing, 1, 161-170

[8] Carnier, R. M., Li, Y., Fujimoto, Y., & Shikata, J. (2024). Deriving Exact Mathematical Models of Malware Based on Random Propagation. Mathematics, 12(6), 835.

[9] Gobinathan, B., Mukunthan, M. A., Surendran, S., Somasundaram, K., Moeed, S. A., Niranjan, P., ... & Sundramurthy, V. P. (2021). A novel method to solve real time security issues in software industry using advanced cryptographic techniques. Scientific Programming, 2021(1), 3611182

[10] Rajalakshmi, M., Saravanan, V., Arunprasad, V., Romero, C. T., Khalaf, O. I., & Karthik, C. (2022). Machine Learning for Modeling and Control of Industrial Clarifier Process. Intelligent Automation & Soft Computing, 32(1).

[11] Praghash, K., Yuvaraj, N., Peter, G., Stonier, A. A., & Priya, R. D. (2022, December). Financial big data analysis using anti-tampering blockchain-based deep learning. In International Conference on Hybrid Intelligent Systems (pp. 1031-1040). Cham: Springer Nature Switzerland.

[12] Alhoraibi,L., Alghazzawi,D., Alhebshi,R., &Rabie,O.B.J. (2023). Physical layer authentication in wireless networks-based machine learning approaches. Sensors, 23(4),1814.

[13] Cicioğlu,M., &Çalhan,A. (2023). MLaR: machine-learning-assisted centralized link-state routing in software-defined-based wireless networks. Neural Computing and Applications, 35(7),5409-5420.

[14] Kaya,O., Güçlüoğlu,T., &İlhan,H. (2024). Depth Analysis in Deep Learning-Based Automatic Modulation Classification. Journal of Aeronautics & Space Technologies/ Havacilikve Uzay Teknolojileri Dergisi,17(2).

[15] Zaeri,N.,& Qasim,R.R.(2023). Intelligent wireless sensor network for gas classification using machine learning. IEEE Systems Journal,17(2),1765-1776.

[16] Jain,A., Mehrotra,T., Sisodia,A., Vishnoi,S., Upadhyay,S., Kumar,A.,...&Illés,Z.(2023).An enhanced self-learning-based clustering scheme for real-time traffic data distribution in wireless networks. Heliyon,9(7).