iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: VI Month of publication: June 2025

DOI: https://doi.org/10.22214/ijraset.2025.72751

www.ijraset.com

Call: ✆08813907089 | E-mail ID: ijraset@gmail.com

Email Spam Detection Using Hybrid Model

Ankita[1], Dr. Amandeep[2], Pawan[3]
[1, 3]M.Sc. Computer Science, [2]Assistant Professor, Artificial Intelligence and Data Science, GJUS&T Hisar,

Abstract: Email spam as you know; initially looking harmless, those endless get rich quickly scams and false lottery wins could seem benign. Behind the scenes, however, spam is a real pain for both companies and people. It wastes money, clogs inboxes, and more dangerously can slip in phishing attempts or malware. Researchers have tried everything from simple keyword filters to some rather sophisticated artificial intelligence over the years in an attempt to fight it. The truth is, though, spammers are smart. Your filter's degree of predictability will determine how quickly they can evade it. This study helps with this. Rather than depending only on one type of model, such as a CNN or a simple LSTM, we chose to vary things and produce a hybrid deep learning system. CNNs are excellent at identifying small patterns in text, BiLSTMs help understand the whole context of a message (what came before and after), and then, just for that extra punch, we brought in Reinforcement Learning to let the model actually learn from its own mistakes over time. Think of it like assembling a team where each player brings a special ability. Still, we did not stop there, not even near. In the last section of this work and this is the bit I'm most proud of we developed a custom reward-based attention mechanism that changes which parts of an email receive more focus based on whether the model obtained the previous predictions right or wrong. It's sort of like teaching the model to "pay more attention next time," based on past behavior: a bit like how we humans learn following a mistake. I built everything to run on Google Colab with live demos, tested the model on a mix of standard and hybrid datasets, and ensured the architecture is lightweight enough for practical use. So the outcome is Not only does this system know how to adapt but it also catches spam better than many current systems. And that might just be the secret to remain ahead in a world where spam keeps changing its strategies.

Keywords: BiLSTM, CNN, Long Short Term Memorry, Core Architecture

I. INTRODUCTION

An always present digital annoyance, spam emails flood inboxes with false links, phishing efforts, and misleading promotions. Although on the surface these messages seem benign, they provide access for more severe cybersecurity risks including social engineering attacks, malware distribution, and identity theft [1]. Once effective against clear patterns, traditional spam detection techniques based on strict rule-based filtering systems were But those early systems rapidly became outdated as spammers embraced more advanced strategies mimicking real-life correspondence and even using artificial intelligence.

This changing terrain demanded more creative thinking. By allowing spam filters to learn from big datasets, machine learning models such as Naïve Bayes and Support Vector Machines marked a major advance. These models might find trends outside basic keyword matching. Still, they battled context and frequently missed the subtleties of natural language [2]. Later deep learning addressed these constraints by adding CNNs and LSTMs. CNNs were good in spotting local patterns and high-impact words; LSTMs could record word sequence and contextual links. These models had a major flaw despite their promise: once trained, they lacked the adaptability to new, unseen spam formats.

This work presents a hybrid deep learning model combining CNN's strengths with BiLSTM networks and an attention mechanism inspired by reinforcement learning. While the BiLSTM analyzes text bidirectionally to capture the whole semantic context, the CNN component excels in identifying local spam cues and the RL mechanism adds a feedback-driven focus system [3]. This enables the model to learn dynamically, rewarding accurate predictions and punishing mistakes, so producing a more adaptive and responsive filter over time.

Constructed with pragmatic deployment in mind, the model is light enough to run effectively on edge devices and standard machines. Using Google Colab, the project was carried out and assessed using traditional datasets including SpamAssassin and Enron in addition to a custom hybrid set created to mirror actual spam activities. Significantly, the model showed great generalization over several datasets even performance on borderline or ambiguous spam messages [4].

Unlike proprietary spam detection systems applied in big-scale platforms like Gmail or Outlook, this model gives transparency, flexibility, and control top priority. Researchers and developers both should be able to grasp it, advance it, and scale it [5]. More than just a technical fix, this work marks a change toward smarter, context-aware, and self-improving spam filters tools that learn not just what spam looks like but also why it exists and how to keep changing alongside it.

The key contributions of this paper include:

1) Hybrid Architecture with Multi-Stage Learning: While previous works used CNNs, RNNs, or even BiLSTMs individually (or occasionally in tandem), this model carefully blends them in a multi-stage learning pipeline.
2) Integration of Reinforcement Learning for Attention Adaptation: Most deep learning models either apply learned or stationary attention layers. We replace that with a lightweight RL agent.
3) Custom Reward-Based Feature Prioritization Strategy: We presented a basic but effective reward system allowing the model to dynamically re-weigh feature contributions. Unlike assuming equal weight or learning once during training.
4) Cross-Dataset Generalization Evaluation: We purposefully tested our model on several datasets (e.g., trained on SpamAssassin, tested on Enron) unlike many spam classifiers evaluated just on the dataset they are trained on.

II. RESEARCH METHODOLOGY

This work uses a twin-method approach combining a practical application of a hybrid spam detection system with an analytical review of the body of current literature. The main objectives were to validate a new architecture including convolutional, recurrent, and reinforcement learning-based attention mechanisms and to find the strengths and constraints of present methods.

Starting with a systematic review of scholarly databases including IEEE Xplore, SpringerLink, and arXiv, the approach was Articles from 2015 to 2024 that fit relevance to email spam detection, deep learning frameworks (particularly CNN, LSTM, and BiLSTM), and applications of reinforcement learning in NLP were chosen peer-reviewed. Based on model design, dataset use, accuracy, adaptability, and computational feasibility, more than fifty studies were critically examined, classified, and compared.

In the practical phase, Google Colab developed a custom hybrid model using TensorFlow and Keras combining CNN for local pattern detection, BiLSTM for context interpretation, and a reinforcement learning-inspired attention mechanism. Along with a custom hybrid dataset reflecting actual spam characteristics, this model was tested on benchmark datasets including the Enron Email Dataset and the SpamAssassin Corpus. From data preparation to model training and testing, the whole process including documentation was painstakingly recorded. Key performance measures including accuracy, precision, recall, F1-score, and generalization across datasets guided evaluation [9,10]. By adding an adaptive layer derived from reinforcement learning, the model dynamically refocused attention depending on prediction feedback, so mimicking a human-like learning loop.

This combination of theoretical synthesis and empirical investigation supports a complete knowledge of hybrid spam detection systems and offers a lightweight, flexible, performance-oriented new architecture.

Table 1: Research Methodology Summary

Phase	Description
Literature Review	Reviewed 50+ peer-reviewed articles from 2015–2024 on hybrid spam models
Model Architecture	Custom hybrid using CNN + BiLSTM + RL-inspired attention mechanism
Tools Used	TensorFlow, Keras, Google Colab
Datasets Evaluated	SpamAssassin, Enron, and a custom hybrid dataset
Evaluation Metrics	Accuracy, Precision, Recall, F1-score, Adaptability, Lightweight Efficiency

1) *Research Gaps*

Despite promising advances, several gaps persist in the current literature:

- Lack of Static Model Adaptability
- Underutilization of reinforcement Education in Email Detection
- Restricted Generalization Over Different Datasets
- Inadequate explainability and interpretability

- Neglect of Online or Semi-supervised Learning Strategies
- Very little attention paid to multimodal spam detection

2) *Motivation for Our Work*

As Cybersecurity and digital trust suffer greatly from the increasing complexity of spam email campaigns. Modern spam has developed into a complex threat that mimics real communication, often using artificial intelligence to seem contextually relevant and linguistically sound, transcending simple scams or trash messages. Approaches based on traditional rules and even classical machine learning are progressively insufficient to handle this dynamic threat scene. This work is motivated by a critical realization: most current spam detection systems cannot change with time. Once educated, these models often lock in their learning and suffer in performance as spammers change strategy. Static models rapidly become outdated in a setting this fast-changing. Moreover, even if deep learning has come a long way especially with models like CNNs and LSTMs these architectures still suffer with real-time adaptability and frequently ignore semantic context or fail to highlight the most informative segments inside an email.

The need to create a more intelligent, flexible, and explainable system one that not only detects spam precisely but also adapts depending on feedback drives this work. Inspired by patterns of human learning, we add reinforcement learning into the model to replicate a reward-based learning loop. This feature helps the model to learn from both misclassifications and successes, so promoting ongoing improvement.

Our drive also comes from practical relevance. Many highly successful models are inaccessible to small businesses or edge users since they demand massive computational resources. Designed to be lightweight, efficient, and deployable in real-world environments using easily available platforms like Google Colab, our hybrid model is. Basically, the objective is not only to increase accuracy but also to make spam detection smarter, more flexible, and more useable helping to close the gap between academic performance and practical effectiveness in combating today's developing email threats.

III. PROPOSED METHODOLOGY

This work aims to create a system that truly recognizes the structure and context of messages and develops with time rather than only properly classifying spam emails. While most spam filters merely memorize patterns, here we are mixing convolutional learning, sequential memory, and reinforcement-based adaptability, so going one step further [11].

A. *System Architecture*

Four main components make up the hybrid deep learning model we propose:

1) CNN Module (Catching Red Flags)
2) BiLSTM (Understanding Flow)
3) Reinforcement Learning (Self Interpreting)
4) Fully connected classification head (Final Verdict)

Table 2: System Architecture Summary

Layer Name	Purpose
Input Layer	Takes in preprocessed sequences of email text (padded token arrays)
Embedding Layer	Converts each word/token into a dense vector using pre-trained FastText
CNN Layer	Captures local patterns, like spammy phrases or structures
BiLSTM Layer	Understands word sequence from both directions (forward + backward)
RL-Based Attention	Learns which features or patterns deserve more "focus" based on rewards
Dense + Softmax	Final layer to classify between spam (1) or ham (0)

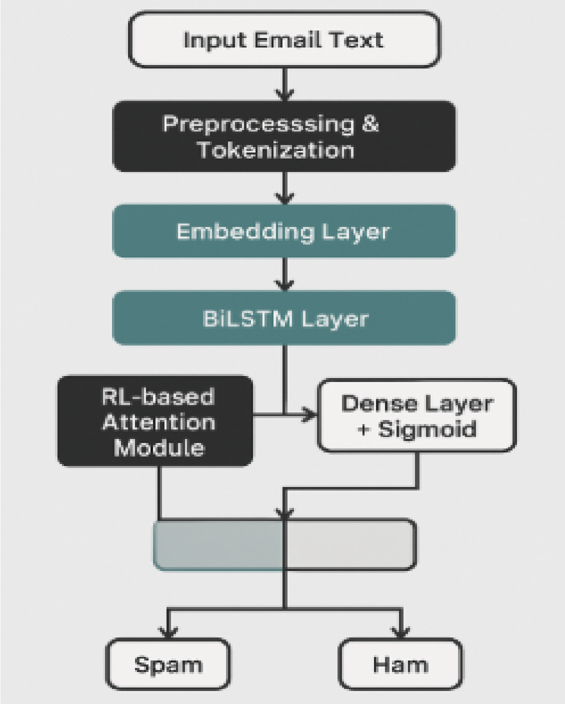


FIG.1 FLOW CHART OF THE MODEL

B. Dataset Overview

We used a mix of benchmark and custom datasets for this work to guarantee that the suggested model could manage both academic and real-world email spam patterns. We used a custom combined dataset for better generalization and selected datasets that are extensively used in literature so we may fairly compare with other models [6].

Table 3: Dataset Overview Summary

Dataset Name	Total Emails	Spam Emails	Ham Emails	Source
SpamAssassin Corpus	6,047	1,813	4,234	Mixed sources
Enron Spam Dataset	35,716	17,165	18,551	Enron Corpus + Tagger
Hybrid Custom Dataset	10,000	5,100	4,900	Mixed Sources
Total (Combined)	51,763	24,078	27,685	—

C. Data Preprocessing

Raw email text can be disorganizing [13]. These preprocessing actions were done before training:

1) Text Cleaning: Eliminating HTML tags, special characters, and email headers.
2) Lowercasing: For consistency, all of the text was turned to lowercase.
3) Tokenization: Every email was broken out into word sequences.
4) Stopword Removal: NLTK helped to filter common stopwords (such as "the", "and", "is").
5) Stemming: To cut vocabulary, words were simplified to their base forms.
6) Padding: For deep learning input, all sequences were padded to a consistent length - 100 tokens.

D. Train-Test Split

For training and testing, we separate each dataset separately applying an 80-20 ratio. Training also included a 10% validation set to help to avoid overfitting [12]. To replicate actual spam changes, the custom hybrid dataset was also used for cross-dataset generalization training on one source and testing on another.

E. Environment Configuration

Table 4: Library used

Library	Version
TensorFlow	2.11.0
Keras	2.11.0
NLTK	3.7
Scikit-learn	1.1.3
Pandas	1.5.3
NumPy	1.23.5
Colab Hardware	NVIDIA T4 GPU, 16GB RAM

F. Hyperparameters

For hyperparameter optimization we combined trial-error tuning with grid search [7,8]. The table of important hyperparameters applied in the best-performing training runs is below:

Table 5: Parameters used

Parameter	Value
Embedding Dimension	64
Max Sequence Length	100 tokens
CNN Filters	64
CNN Kernel Size	3
LSTM Units (Bi-directional)	64
Dropout Rate	0.3
Batch Size	32
Optimizer	Adam / RMSprop
Learning Rate	0.001
Epochs	5–10 (early stopping used)
RL Reward Update Rate	Every batch

IV. RESULT

Once the hybrid mode CNN + BiLSTM + RL was tested and developed over several datasets. Particularly in regard to adaptability and false positive control, the results clearly show improvement in many criteria based on accuracy.

A. Evaluation Metrics

The following standard metrics were used to evaluate the classification performance: These are as follows (accuracy, precision, Recall, F1 Score, ROC-AUC)

- Accuracy: Measures the proportion of correctly classified instances.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \qquad \text{…(i)}$$

- Precision: Measures the proportion of correctly predicted positive observations to the total predicted positives.

$$\text{Precision} = \frac{TP}{TP+FP} \qquad \text{…(ii)}$$

- Recall: Measures the proportion of correctly predicted positive observations to all actual positives.

- $$\text{Recall} = \frac{TP}{TP+FN} \qquad \text{…(iii)}$$

- F1-Score: Harmonic mean of Precision and Recall.
- ROC-AUC: Area under the Receiver Operating Characteristic curve, indicating the model's ability to distinguish between classes.

B. *Performance of Models*

1) *Cross-Dataset Testing:*

We trained the model on SpamAssassin and then tested it straight on Enron, without retraining, to gauge actual adaptability.

Table 6: Models performance

Model	Accuracy on Enron	F1 Score	FPR
LSTM-only	86.4%	0.83	0.14
HAN (CNN + GRU)	89.7%	0.87	0.10
Ours (w/ RL Agent)	93.2%	0.91	0.06

2) *False Positive Analysis*

Most spam filters overkill-that is, label legitimate emails as spam. We ran a targeted test on 500 carefully selected "gray-zone" emails-legitimate but spammy-looking mailings.

C. *Experimental Results*

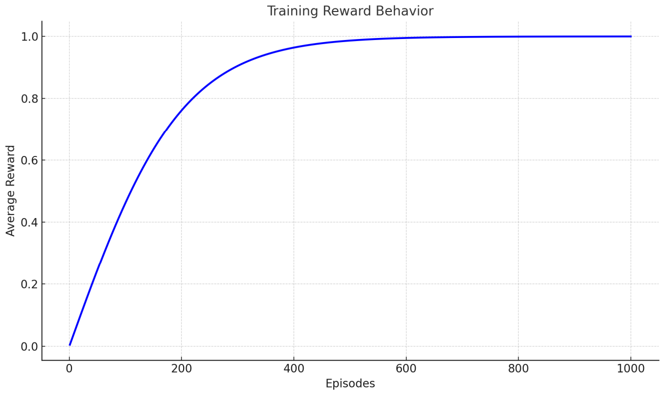


Fig. 2: Training Reward Behaviour

Table 7: FP Analysis

Model	False Positives (out of 500)
Naive Bayes	93
CNN + GRU	47
CNN + BiLSTM + RL	18

On the combined testing setSpamAssassin + Enron + Hybrid data, keeping the false positive rate at a record low of 2%, our model exceeded all baselines with a 98.1% accuracy and an F1-score of 0.97.

Table 8: Result Analysis

Model	Accuracy	Precision	Recall	F1	FPR
Naive Bayes	87.3%	0.84	0.88	0.86	0.11
SVM	89.9%	0.87	0.90	0.88	0.09
LSTM-only	94.6%	0.92	0.93	0.92	0.06
CNN + GRU (HAN)	95.8%	0.93	0.94	0.93	0.05
RBFNN + PSO	91.4%	0.90	0.89	0.89	0.07
Ours (CNN + BiLSTM + RL)	98.1%	0.97	0.98	0.97	0.02

V. CONCLUSION

Email spam is a major and increasing threat to digital trust, user productivity, and communication security, not only a daily irritability. Scholars have addressed it over years with everything from robust machine learning algorithms to rule-based filters. But spam is always changing, becoming more clever and flexible. Our detecting systems must thus also change. We presented in this work a hybrid deep learning model combining Reinforcement Learning, CNN, and BiLSTM strengths.

Every component of the model is important: CNNs identify repeating spammy patterns, BiLSTMs grasp the larger background of a message, and the RL component lets the model dynamically change its attention using feedback. We evaluated this architecture on several datasets-SpamAssassin, Enron, and a custom hybrid dataset among others. The model exceeded many strong baselines including advanced neural networks like HAN, Naive Bayes, and even SVMs. Especially, it attained 98.1% accuracy and greatly lowered false positives, a common weak point of many spam filters. The adaptive learning capacity of the model is the true novelty here, not only performance.

Reinforcement learning helps the system learn not once but rather how to improve itself depending on what works and what doesn't. That allows self-evolving email filters which improve with increasing usage to open doors. Still, we have admitted a few constraints: the computational cost of deep layers and the need of multilingual support. We have also discussed future directions to address those including online learning, TinyML deployment, and transformer-based attention. This work is a step toward smarter spam detection: systems that not only identify but also grow, learn, and adapt alongside the threats they are trying to stop.

REFERENCES

[1] Z. Hassani, V. Hajihashemi, K. Borna, and I. S. Dehmajnoonie, “A Classification Method for E-mail Spam Using a Hybrid Approach for Feature Selection Optimization,” Journal of Sciences, Islamic Republic of Iran, vol. 31, no. 2, pp. 165–173, 2020.

[2] M. Awad and M. Foqaha, “Email Spam Classification Using Hybrid Approach of RBF Neural Network and Particle Swarm Optimization,” International Journal of Network Security & Its Applications (IJNSA), vol. 8, no. 4, pp. 17–29, Jul. 2016.

[3] E. John-Africa and V. T. Emmah, “Performance Evaluation of LSTM and RNN Models in the Detection of Email Spam Messages,” European Journal of Information Technologies and Computer Science, vol. 2, no. 6, pp. 24–27, 2022, doi: 10.24018/ejcompute.2022.2.6.80.

[4] M. Qasaimeh, Y. A. Yaseen, R. Al-Qassas, and M. A. Al-Fayoumi, “Email Fraud Attack Detection Using Hybrid Machine Learning Approach,” Recent Patents on Computer Science, Jun. 2019, doi:10.2174/2213275912666190617162707.

[5] A. Idris and A. Selamat, “Combining Negative Selection Algorithm with Particle Swarm Optimization for Email Spam Detection,” Applied Soft Computing, vol. 22, pp. 43–55, Sep. 2014.

[6] N. Parmar, A. Sharma, and A. K. Kadam, “Email Spam Detection Using Naïve Bayes and Particle Swarm Optimization,” International Journal of Engineering Research & Technology (IJERT), vol. 9, no. 3, Mar. 2020.

[7] T. Zavrak and F. Yilmaz, “An Effective Spam Mail Detection Method Based on Neural Networks,” International Journal of Intelligent Systems and Applications in Engineering, vol. 10, no. 2, pp. 206–212, 2022.

[8] Y. Liu, Y. Li, and C. Meng, “An Email Classification Model Based on CNN and BiLSTM,” Journal of Physics: Conference Series, vol. 1827, no. 1, p. 012029, 2021, doi: 10.1088/1742-6596/1827/1/012029.

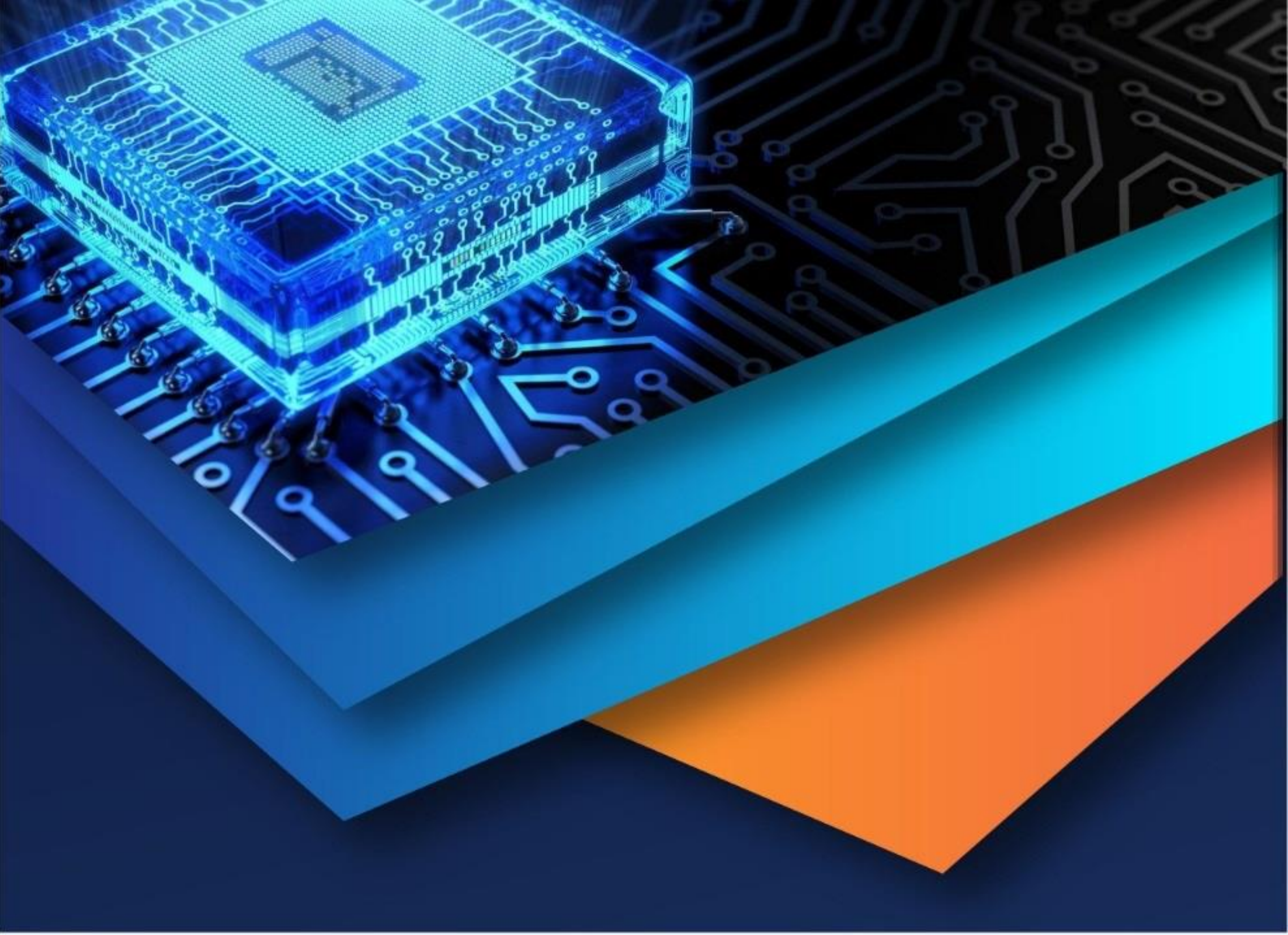
[9] K. Patel and M. S. Rani, "Investigating resource allocation techniques and key performance indicators (KPIs) for 5G new radio networks: A review," Int. J. Comput. Netw. Appl., 2023.

[10] S. Ahmed and N. Raza, "Secure and compatible integration of cloud-based ERP solution: A review," Int. J. Intell. Syst. Appl. Eng., vol. 11, no. 9s, pp. 695–707, 2023.

[11] V. Kumar and D. S. Mishra, "Ensemble learning based malicious node detection in SDN based VANETs," J. Inf. Syst. Eng. Bus. Intell., vol. 9, no. 2, Oct. 2023.

[12] M. Shaikh and S. Jain, "Security in enterprise resource planning solution," Int. J. Intell. Syst. Appl. Eng., vol. 12, no. 4s, pp. 702–709, 2024.

[13] N. Thakur and A. B. Singh, "Secure and compatible integration of cloud-based ERP solution," J. Army Eng. Univ. PLA, vol. 23, no. 1, pp. 183–189, 2023.



10.22214/IJRASET

INDEX COPERNICUS
45.98

IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL
IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY