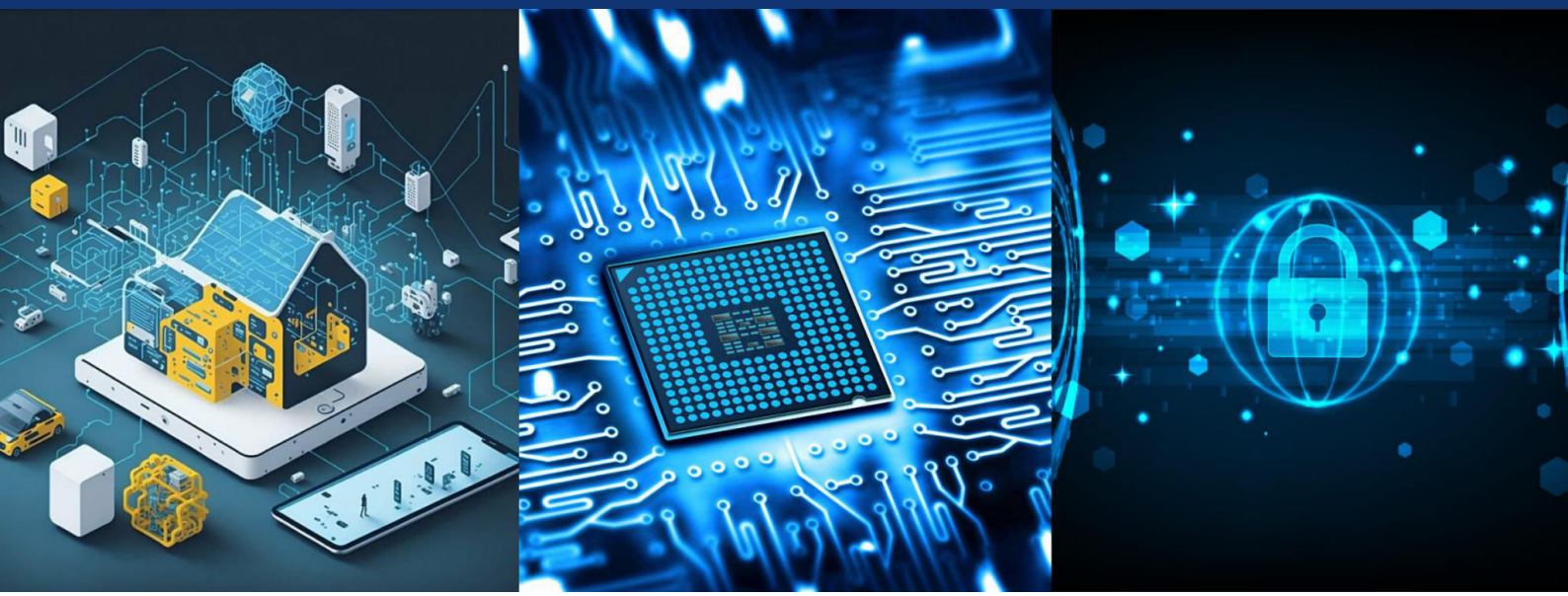
IJIRCCE

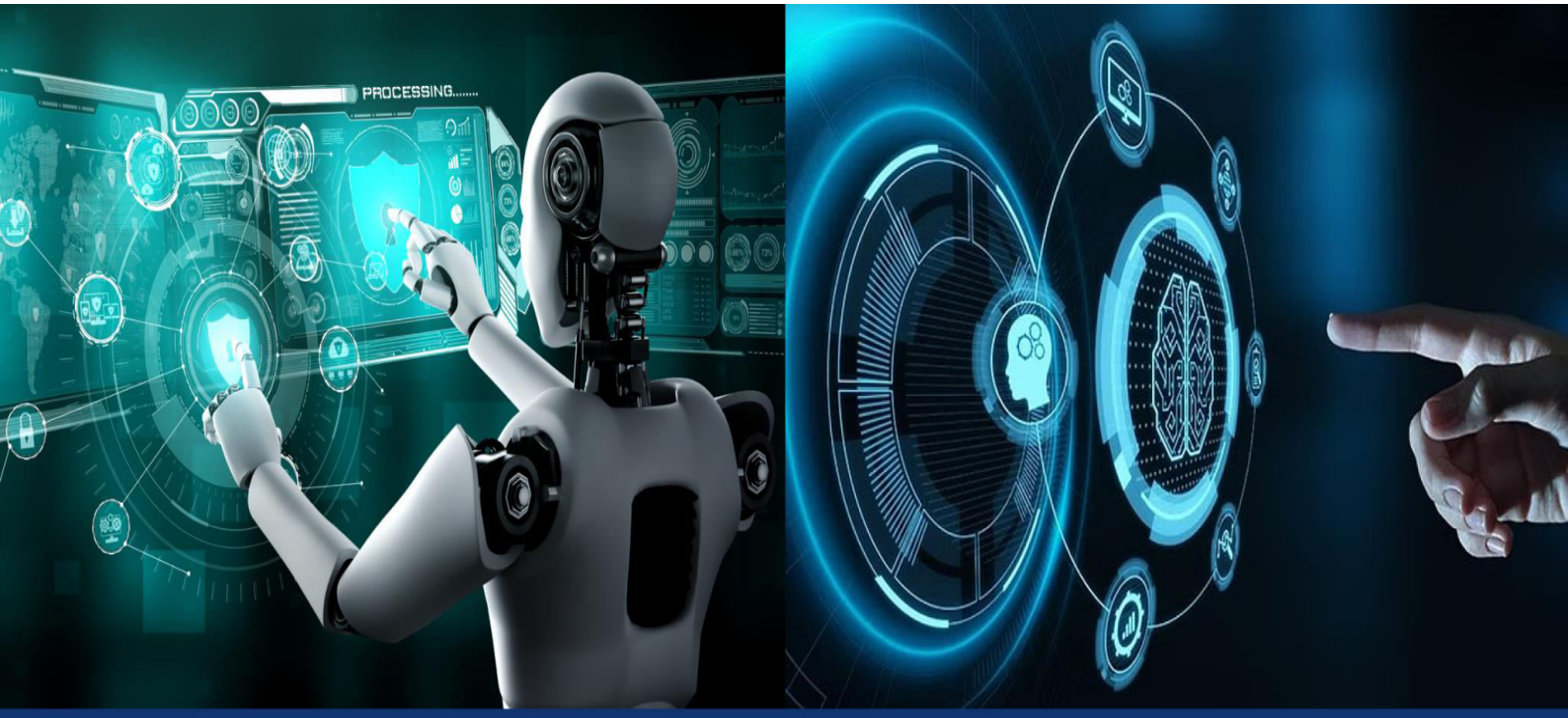
ISSN(O): 2320-9801

ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.927

Volume 14, Issue 7, July 2026

© 2026 IJIRCCE | Volume 14, Issue 7, July 2026 DOI: 10.15680/IJIRCCE.2026.1407018

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Enhanced Real Time Detection of Polymorphic Malware Using Synthetic Data Augmentation

Anisetti Amitha, Dr. M. Nagaratna

Post-Graduate Student, Cyber Forensics and Information Security, Department of Computer Science Engineering, Jawaharlal Nehru Technological University, Hyderabad, India

Professor, Department of Computer Science Engineering, Jawaharlal Nehru Technological University, Hyderabad, India

ABSTRACT: The evolution of malware led to the emergence of polymorphic variants capable of continuously modifying their code structure to evade traditional detection mechanisms. Such adaptability renders conventional signature-based and heuristic approaches largely ineffective, creating a need for intelligent and adaptive security solutions. This research presents a framework for real-time detection of polymorphic malware, integrating deep learning and synthetic data augmentation to enhance robustness and generalization. The proposed system leverages Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Attention LSTM along with other algorithms like Random Forest, Voting Classifier, Stacking Classifier and Naive Bayes to jointly analyze both static and dynamic features of malware, enabling the detection of complex behavioral patterns. To address the challenge of limited labeled datasets, a Generative Adversarial Network (GAN) is employed to generate synthetic data, thereby enriching the training samples and improving model diversity. The framework incorporates a data processing pipeline, including feature engineering, balancing, and real-time traffic analysis. To enhance transparency and trust, Explainable AI techniques such as LIME and SHAP are employed to interpret feature contributions and model decisions. Furthermore, a Flask-based web application is developed to enable real-time malware prediction through a user-friendly interface

KEYWORDS: Polymorphic Malware,Cybersecurity, Machine Learning, Deep Learning, CNN, RNN, Attention-LSTM, GAN, Synthetic Data Augmentation, Explainable AI, Real-Time Malware Detection.

I. INTRODUCTION

The rapid growth of digital technologies, cloud computing, Internet of Things (IoT) devices, and inter-connected systems have exponentially increased the volume and complexity of cyber threats. Malware remains one of the most significant threats to cybersecurity. Malware is a collective term used to describe malicious software that is designed to harm computer systems, steal sensitive data, disrupt services, or gain unauthorized access to networks. With the changing nature of cyberattacks, traditional security measures are finding it increasingly difficult to defend against more complex malware strains.Signature-based detection is the most common technique in use today, where files or network traffic are compared against a database of known malware signatures. These approaches are helpful to detect known threats but not useful against zero-day attacks and newly found malware.

Heuristic based detection methods offer some benefits by detecting suspicious features, but often they have high false-positive rates and low flexibility.Polymorphic malware is becoming one of the most difficult kinds of malware, because it can constantly change its code structure while preserving its destructive activity.

Fast code changes are making it hard to detect early as it can escape traditional signature-based detection techniques. Thus, behavior-based detection approaches relying on machine learning (ML) and deep learning (DL) have attracted considerable interest due to their ability to detect previously unseen malware variants and uncover complex patterns directly from data.

Insufficient samples for specific attack categories are frequently found in malware datasets, which results in biased model training and poorer generalization performance. Generative Adversarial Networks (GANs) can be used to create realistic synthetic malware samples in order to solve this problem. Synthetic data augmentation promotes dataset diversity, eliminates class imbalance, and improves the robustness of machine learning and deep learning models.

© 2026 IJIRCCE | Volume 14, Issue 7, July 2026 DOI: 10.15680/IJIRCCE.2026.1407018

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

This study suggests a methodology for real-time polymorphic malware detection that combines explainable artificial intelligence, machine learning, deep learning, and synthetic data augmentation. To efficiently evaluate malware activity, the framework makes use of CNN, RNN, Attention-LSTM, Random Forest, Voting Classifier, Stacking Classifier, and Naive Bayes models. While LIME and SHAP improve prediction interpretability, GAN-based synthetic data generation is used to improve dataset quality and model performance. Additionally, a Flask-based web application that provides real-time virus prediction through an interactive and user-friendly interface is used to implement the suggested approach. In contemporary cybersecurity settings, the suggested approach seeks to offer a precise, scalable, and trustworthy way to identify emerging polymorphic malware.

II. RELATED WORK

By making it possible to recognize intricate behavioral patterns that are challenging to identify with traditional security measures, recent developments in artificial intelligence (AI) have greatly enhanced malware detection. The authors of [1] suggested a deep learning system that combines deep neural network designs with behavioral feature extraction to identify polymorphic malware. Their methodology showed enhanced detection accuracy against changing ransomware strains and demonstrated how deep learning can be used to find threats that have never been observed before. However, the study did not take into account numerous malware families or real-time deployment and instead concentrated on ransomware detection.

In [2], an AI-driven framework for polymorphic malware detection was introduced. Rather than using conventional signature-based techniques, machine learning techniques were used to study malware behavior. By learning behavioral traits from gathered datasets, the suggested approach enhanced the detection of constantly evolving malware. The approach did not address the issues of dataset imbalance, explainable predictions, or deployment in real-time operating situations, despite achieving impressive detection performance.

In [3], a thorough survey of polymorphic malware analysis was carried out, examining current machine learning and deep learning methods as well as static, dynamic, and hybrid malware detection strategies. The report highlighted the growing significance of intelligent detection techniques while discussing the difficulties presented by code obfuscation, encryption, metamorphism, and dataset limits. The authors recognized the need for more resilient frameworks with enhanced data availability and sophisticated learning algorithms to manage changing virus behaviors.

Real-time malware forensic analysis utilizing Artificial Intelligence was examined in [4], where AI-based models were applied to automate malware investigation and behavioral analysis. By minimizing the need for manual analysis, the suggested methodology helped security analysts in forensic investigations and made it possible to identify malicious activity more quickly. Although the method increased the effectiveness of investigations, it continued to prioritize malware forensics above thorough real-time malware classification and prediction.

III. PROPOSED METHODOLOGY

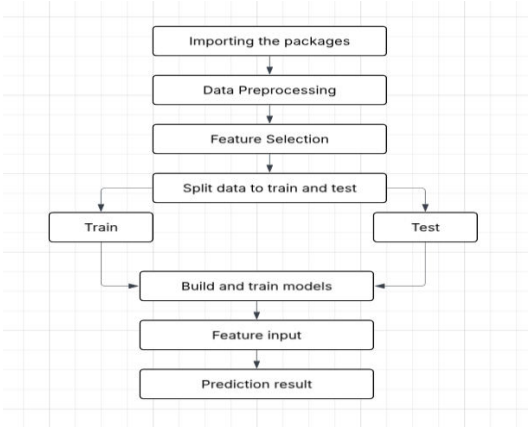


Fig. 1. Architecture

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

A. *System Architecture*

The architecture of the suggested Enhanced Real-Time Detection Framework for Polymorphic Malware Using Synthetic Data Augmentation is shown in Figure 1. The framework combines explainable artificial intelligence (XAI), machine learning, deep learning, ensemble learning, data preparation, synthetic data generation, and real-time deployment into a single malware detection system.

The first step in the process is gathering malware traffic data that includes several malware families, such as Trojan, Worm, Ransomware, Spyware, Benign, and Botnet. To enhance data quality and consistency, the gathered dataset is put through a thorough preprocessing step that includes data cleaning, missing value handling, duplicate removal, categorical feature encoding, feature scaling, and feature engineering. SMOTE (Synthetic Minority Oversampling Technique) is initially used to balance minority malware classes because malware datasets typically include class imbalance. To further boost dataset variety and enhance model generalization, a Generative Adversarial Network (GAN) then creates realistic synthetic malware samples.

After the dataset has been analyzed, redundant attributes are removed and the most informative features are chosen. Several machine learning and deep learning classifiers, such as Random Forest, Naive Bayes, CNN, RNN, Attention-LSTM, Voting Classifier, and Stacking Classifier, receive the optimal feature subset as input. Each classifier predicts malware categories and learns malware behavioral patterns on its own.

Accuracy, Precision, Recall, F1-Score, ROC-AUC Score, and Confusion Matrix are used to assess each model's performance. The Stacking Classifier is chosen as the final malware detection model because it achieves the best overall performance based on comparative evaluation.

LIME and SHAP are incorporated into the framework to clarify prediction choices and determine each feature's contribution to malware categorization in order to increase model transparency. Lastly, a Flask-based web application is used to deploy the trained Stacking model, enabling users to input malware traffic features and receive real-time malware predictions along with confidence scores and explainable AI visuals.

B. Data Preprocessing and Synthetic Data Generation

Prior to model training, the suggested architecture uses a methodical preprocessing pathway to enhance data quality. By creating artificial minority samples, SMOTE is used to balance minority virus classes.Two neural networks make up the GAN:

The Generator creates artificial malware samples from random noise.

$$G(z) \rightarrow xfake$$

where

- z represents random latent noise,
- $xfake$ represents generated malware samples.

Real and fake malware samples are distinguished by the Discriminator. To increase training variability and lessen overfitting, the synthetic samples produced by GAN are combined with the original dataset.

C. Malware Classification

Several machine learning and deep learning classifiers are given access to the optimal dataset following preprocessing and feature selection.

The malware category is predicted by each classifier.

$$\hat{y} = f(x)$$

where

- x denotes the feature vector,
- $\hat{y}$ denotes the predicted malware label.

The proposed framework trains the following models:

- Random Forest
- Naive Bayes
- CNN
- RNN

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

- Attention-LSTM
- Voting Classifier
- Stacking Classifier

A number of performance measures are used to evaluate each model separately. Predictions from several base learners are combined by the Stacking Classifier,

$$S(x) = fmeta(f1(x), f2(x), \ldots, fn(x))$$

where

- $f1, f2, \ldots, fn$ denote the base classifiers,
- $fmeta$ represents the meta-classifier.

The classifier producing the highest validation accuracy is selected for deployment.

D. Explainable AI

Even though the suggested framework produces excellent detection accuracy, it is just as crucial to comprehend prediction conclusions. Thus, LIME and SHAP are integrated in the suggested system.

SHAP uses Shapley values to calculate each feature's contribution, while LIME uses an interpretable model to estimate the prediction locally.

E. Real-Time Malware Prediction

The trained Stacking Classifier is deployed through a Flask web application.

The system displays

- Malware Category
- Prediction Confidence
- SHAP Explanation
- LIME Explanation

allowing real-time malware detection through a user-friendly interface.

IV. PSEUDO CODE

Algorithm: Enhanced Real-Time Detection of Polymorphic Malware

Step 1: Gather statistics about malware traffic.
Step 2: Complete preprocessing by encoding categorical attributes, eliminate duplicate data and missing values.
Step 3: Conduct feature engineering and normalize numerical features.
Step 4: Apply SMOTE to balance minority malware classes.
Step 5: Use the balanced dataset to train the GAN.
Step 6: Use the trained Generator to create fake malware samples.
Step 7: Combine datasets of original and generated malware. .
Step 8: Choose features.
Step 9: Split the dataset into training and testing sets.
Step 10: Train Random Forest classifier.
Step 11: Train Naive Bayes classifier.
Step 12: Train CNN model.
Step 13: Train RNN model.
Step 14: Train Attention-LSTM model.
Step 15: Train Voting Classifier.
Step 16: Train Stacking Classifier.
Step 17: Evaluate all classifiers using Accuracy, Precision, F1-Score and Confusion Matrix.
Step 18: Select the classifier with the highest performance (Stacking Classifier).
Step 19: Generate prediction explanations using SHAP and LIME.
Step 20: Deploy the trained Stacking model using a Flask web application.
Step 21: Accept user input, predict the malware class, display prediction confidence and XAI explanations.
Step 22: End.

© 2026 IJIRCCE | Volume 14, Issue 7, July 2026 DOI: 10.15680/IJIRCCE.2026.1407018

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

V. SIMULATION RESULTS

The CyberTec IIoT Malware Dataset, which comprises benign, botnet, ransomware, spyware, Trojan, and worm traffic, was used to assess the suggested Enhanced Real-Time Detection Framework for Polymorphic Malware Using Synthetic Data Augmentation. After preprocessing the dataset with data cleaning, label encoding, feature scaling, feature selection, and SMOTE-based balancing, GAN-based synthetic data creation was used to increase dataset diversity. Naive Bayes, Random Forest, CNN, RNN, Attention-LSTM, CNN-RNN, CNN-Attention LSTM, Voting Classifier, and Stacking Classifier were used to assess the framework. The efficacy of ensemble learning and synthetic data augmentation for polymorphic malware detection was demonstrated by evaluating the model's performance using Accuracy, Precision, Recall, and F1-Score.

A. Accuracy Analysis

One of the most crucial performance indicators for assessing malware detection programs is classification accuracy. The accuracy of Naive Bayes, Random Forest, CNN, RNN, Attention-LSTM, CNN-RNN, CNN-Attention LSTM, Voting Classifier, and Stacking Classifier is compared in Table I.

The Stacking Classifier had the highest accuracy of 91.8%, followed by the Voting Classifier with 90.2% and the Random Forest classifier with 89.3%, as Table I illustrates. With an accuracy of 70.9%, the CNN-RNN hybrid model outperformed CNN, RNN, and Attention-LSTM separately among the deep learning models. The Naive Bayes classifier had the lowest accuracy of 33.4%, suggesting that complicated polymorphic malware activity is harder to identify using simple probabilistic classification.

B. Precision Analysis

Precision quantifies the classifier's capacity to accurately detect malicious traffic while reducing false positive predictions. A high degree of precision suggests that the anticipated malware samples are quite trustworthy.

The results of the experiment show that the Stacking Classifier had the highest precision of 91.9%, closely followed by Random Forest (89.4%) and Voting Classifier (90.3%). With a precision of 70.8%, the hybrid CNN-RNN model outperformed the separate deep learning architectures. Because of its high independence assumptions, which fail to reflect the intricate linkages found in malware traffic data, the Naive Bayes classifier had the lowest precision of 34.5%. These findings show that as compared to individual machine learning and deep learning models, ensemble learning greatly increases prediction reliability.

C. Recall Analysis

Recall assesses how well the suggested framework can recognize real malware cases. Better detection of harmful samples while lowering false negatives is indicated by a greater recall value.

The suggested Stacking Classifier demonstrated its efficacy in detecting malware traffic across various attack categories with the greatest recall of 91.8%. Recall values for the Random Forest and Voting Classifier were 89.3% and 90.2%, respectively. While CNN, RNN, and Attention-LSTM achieved somewhat lower recall values, the CNN-RNN hybrid model scored a recall of 70.9%. With the lowest recall of 33.4%, the Naive Bayes classifier demonstrated a poor capacity to identify polymorphic malware. The suggested ensemble framework's high recall shows that it can efficiently identify malicious activity while reducing missed detections.

D. F1-Score Analysis

By taking into account both precision and recall at the same time, the F1-Score offers a fair assessment. It is very helpful for malware identification issues with imbalanced datasets and several classes.

The Stacking Classifier has the highest F1-Score (91.8%), followed by the Voting Classifier (90.2%) and Random Forest (89.3%), as shown in Table I. The CNN-RNN architecture yielded the greatest F1-Score of 70.5% among the deep learning models, demonstrating enhanced classification ability through integrated temporal and spatial feature learning. The Naive Bayes classifier performed relatively poorly in detecting polymorphic malware, as seen by its lowest F1-Score of 28.1%. The Stacking Classifier's balanced ability to maximize malware identification while minimizing false predictions is demonstrated by its strong F1-Score.

© 2026 IJIRCCE | Volume 14, Issue 7, July 2026 DOI: 10.15680/IJIRCCE.2026.1407018

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

E. Comparative Performance Analysis

It is evident from the overall experimental comparison that ensemble learning models perform better than both solo deep learning models and conventional machine learning models. SMOTE and GAN-based synthetic data augmentation greatly increased dataset diversity, decreased class imbalance, and improved the models' capacity to generalize to polymorphic malware strains that had never been seen before. All malware categories saw an improvement in prediction performance as a result of the classifiers' ability to learn deeper malware features thanks to the created synthetic examples.

With an accuracy of 91.8%, precision of 91.9%, recall of 91.8%, and F1-score of 91.8%, the Stacking Classifier continuously generated the best overall performance among the classifiers that were assessed. While the hybrid CNN-RNN model much outperformed the individual CNN, RNN, and Attention-LSTM architectures, the Voting Classifier and Random Forest also showed good classification performance.

)ut[60]:

	ML Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
0	RandomForest	0.893	0.894	0.893	0.893	0.986
1	GaussianNB	0.334	0.345	0.334	0.281	0.678
2	RNN	0.521	0.509	0.521	0.509	0.835
3	CNN	0.591	0.583	0.591	0.578	0.878
4	Hybrid_CNN_RNN	0.709	0.708	0.709	0.705	0.925
5	Voting Classifier	0.902	0.903	0.902	0.902	0.989
6	Stacking Classifier	0.918	0.919	0.918	0.918	0.993
7	Attention LSTM	0.549	0.537	0.549	0.533	0.852
8	CNN + Attention LSTM	0.535	0.525	0.535	0.521	0.849

Fig. 2. Comparison results

```
Classification Report:
              precision    recall  f1-score   support

           0       0.81      0.88      0.84      1162
           1       0.92      0.90      0.91      1162
           2       0.97      0.97      0.97      1162
           3       0.92      0.89      0.90      1162
           4       0.93      0.93      0.93      1162
           5       0.96      0.94      0.95      1162

    accuracy                           0.92      6972
   macro avg       0.92      0.92      0.92      6972
weighted avg       0.92      0.92      0.92      6972
```

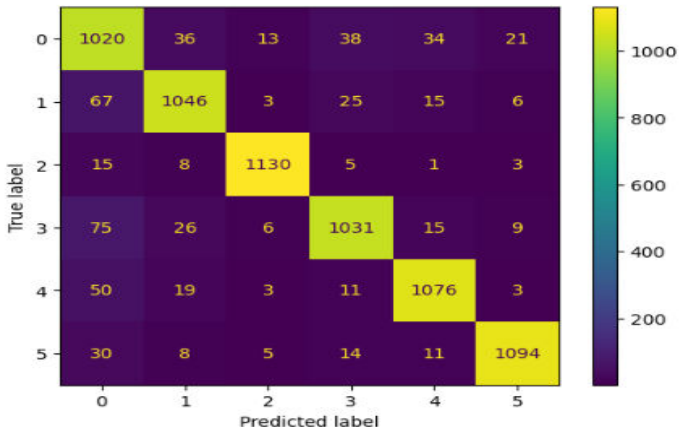


Fig. 3. Stacking Classifier report

© 2026 IJIRCCE | Volume 14, Issue 7, July 2026 DOI: 10.15680/IJIRCCE.2026.1407018

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

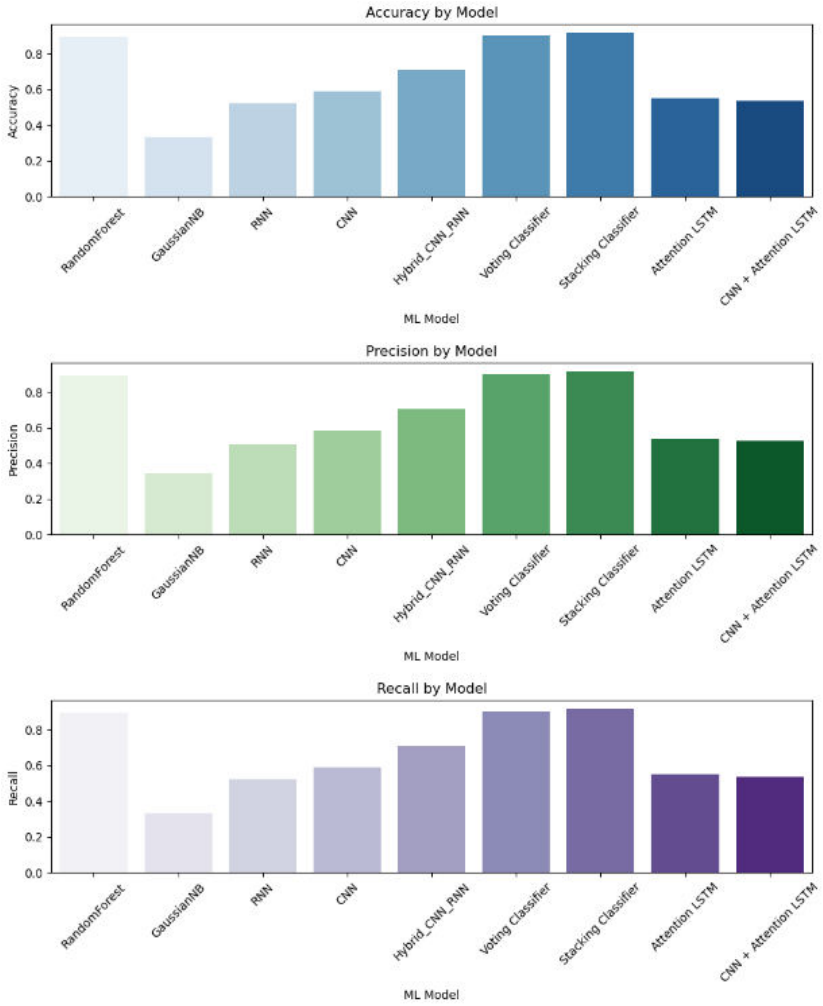


Fig. 4. Accuracy, Precision and Recall

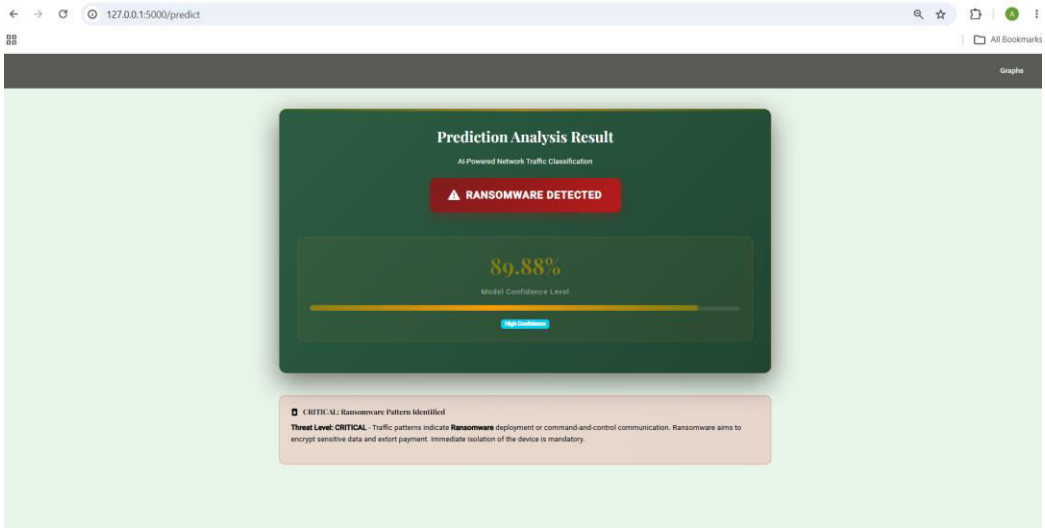


Fig. 5. Malware Detection

© 2026 IJIRCCE | Volume 14, Issue 7, July 2026 DOI: 10.15680/IJIRCCE.2026.1407018

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

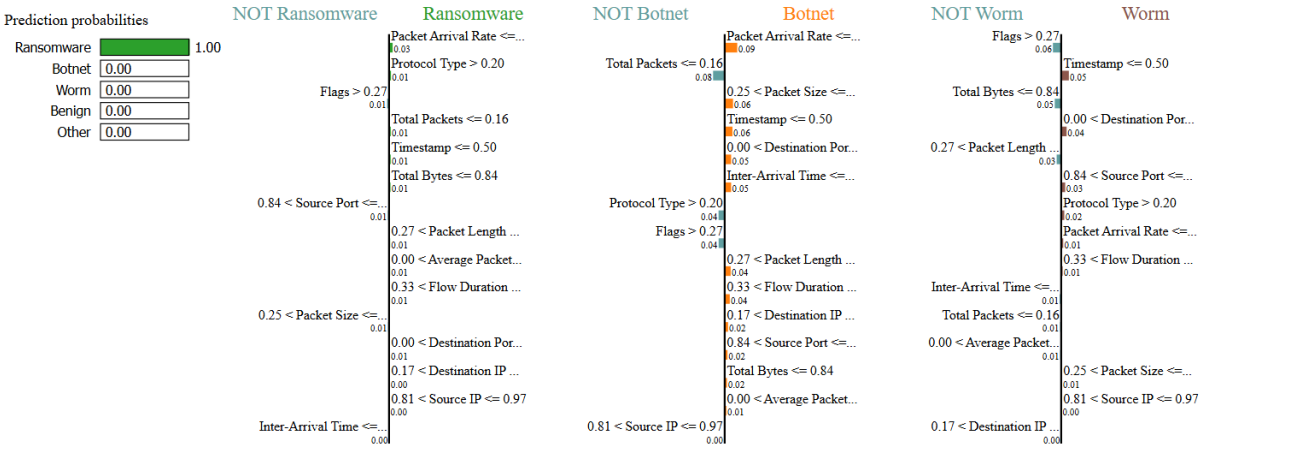


Fig. 6. XAI LIME

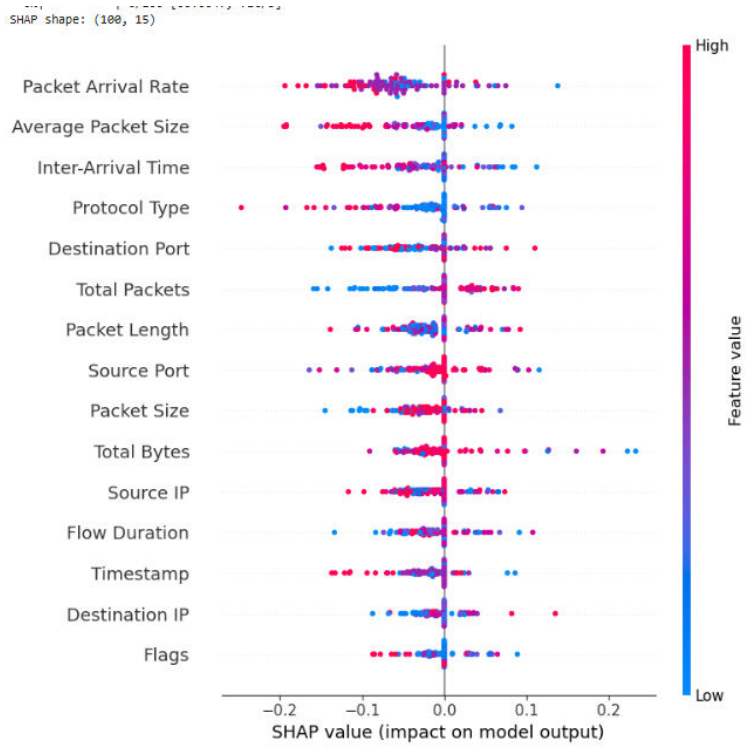


Fig. 7. XAI SHAP

VI. CONCLUSION AND FUTURE WORK

The proposed Enhanced Real-Time Detection Framework for Polymorphic Malware Using Synthetic Data Augmentation approach suggests that by combining machine learning, deep learning, ensemble learning, GAN-based synthetic data augmentation, and Explainable AI, the suggested Enhanced Real-Time Detection Framework for Polymorphic Malware Using Synthetic Data Augmentation successfully enhances malware detection in IIoT environments. The Stacking Classifier outperformed the other models in identifying developing malware variants, with the greatest accuracy of 91.8%. Real-time virus prediction was made possible by the Flask-based web application, while LIME and SHAP integration improved model interpretability. To further increase detection accuracy, adaptability, and scalability, future

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

research will concentrate on integrating sophisticated generative models, online learning strategies, graph-based feature representations, and assessment on bigger real-world IIoT datasets.

REFERENCES

1. M. Gazzan, B. Alobaywi, M. Almutairi and F. T. Sheldon, *'A Deep Learning Framework for Enhanced Detection of Polymorphic Ransomware'*, Future Internet, Vol. 17, Issue 7, Article 311, 2025.
2. M. Anderson, S. Rahman, H. Tanaka, C. Dominguez and L. Yue, *'AI-Based Detection of Polymorphic Malware'*, 2025.
3. M. S. Avhankar, J. Pawar and V. Kumbhar, *'A Comprehensive Survey on Polymorphic Malware Analysis: Challenges, Techniques, and Future Directions'*, 2025.
4. J. Whitman, A. El-Karim, P. Nandakumar, F. Ortega and L. Zheng, *'Real-Time Malware Forensics Using AI'*, 2025.
5. H. C. B. Hamadjida, A. T. Kouanou, C. T. Tchito and C. T. Kouadjo, *'Malware's Polymorphism Analysis Using a Hybrid Machine Learning Algorithm Approach'*, Proceedings of the Data Science Workshop: *From Theory to Practice*, 2025.
6. K. R. Ahmed, M. M. A. Semi, S. Akther, M. M. K. Rabbi, M. R. Raja, U. Chakraborty and M. I. H. Rial, *'Blockchain-Integrated Malware Detection Systems: Enhancing Accuracy and Trust in Cybersecurity'*, Proceedings of the 4th OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 5.0, IEEE, pp. 1–6, 2025.
7. L. Mauri and E. Damiani, *'Hardening Behavioral Classifiers Against Polymorphic Malware: An Ensemble Approach Based on Minority Report'*, Information Sciences, Vol. 689, Article 121499, 2025.
8. S. Sultana, M. Uddin, M. A. R. Chy, S. N. Hasan, E. Hossain, H. Kaur and J. Kaur, *'AI-Augmented Big Data Analytics for Real-Time Cyber Attack Detection and Proactive Threat Mitigation'*, International Journal of Computational and Experimental Science and Engineering, Vol. 11, Issue 3, pp. 5639–5647, 2025.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details