Grenze International Journal of Engineering and Technology, January Issue

An Ensemble Learning Approach for Detecting Steganographic Malware in Digital Images

Savita Adhav[1], Ashish Sarpate[2], Samrat Gaisamudre[3], Sakshi Tirmanwar[4], Smitanshu Ukey[5] and Ganesh Arbad[6]
[1-6]Dept. of Information Technology, Rajarshi Shahu College of Engineering Pune, India
Email: sadhavit@jspmrscoe.edu.in, sarpateashish1@gmail.com, samratgaisamudre1717@gmail.com, tirmanwarsakshi@gmail.com, smitanshuukey3070@gmail.com, arbadg@gmail.com

***Abstract*— Steganography is a growing threat to cybersecurity, as it is employed by steegomalware to conceal harmful payloads within apparently innocuous digital images and evade detection by conventional methods. The sophisticated ensemble-based approach presented in this study is intended to accurately identify and categorize steganographic malware. We incorporate a multiphase workflow that includes hybrid classification, optimization, and feature extraction.**

Initially, incoming photos are processed by a feature extractor to detect any steganographic artifacts. Then, to improve detection robustness, a group of classifiers, such as Random Forest, XGBoost, and Stochastic Gradient Descent (SGDC), operate in tandem with neural network models, such as Convolutional Neural Networks (CNN), Convolutional Autoencoders, and Generative Adversarial Networks (GAN). By using soft voting to compile predictions, the method ensures accurate stegomalware categorization.

Results from experiments show how successful the framework is, finding hidden payloads with almost flawless accuracy, even when they are obfuscated using compression or encoding methods like Base64. Our method fills important holes in current steganalysis tools by fusing the advantages of deep learning and machine learning, providing a scalable and effective defense against changing steganographic threats. By offering a unique, hybrid approach to reducing stegomalware risks in diverse digital environments, this work advances the field of cybersecurity and machine learning.

***Index Terms*— Steganographic Malware, Deep Learning, Ensemble learning, Convolutional Neural Networks, Real-time Detection.**

I. INTRODUCTION

The rapid evolution of cyber threats has led to the emergence of sophisticated malware that leverages steganography—stegomalware—to conceal malicious payloads within seemingly innocuous digital images, documents, or videos [1]. Unlike traditional malware, stegomalware evades signaturebased detection systems by embedding harmful code in the least significant bits (LSB) of image pixels or other covert data channels [2]. This technique allows attackers to bypass conventional security measures, making stegomalware a significant threat to individuals, enterprises, and critical infrastructure [3]. Recent studies highlight the growing prevalence of stegomalware in cyberattacks, including ransomware, spyware, and advanced persistent threats (APTs) [4]. For example, malware families such as AdGholas, Cerber, and Stegano have been documented embedding malicious JavaScript or executable scripts within PNG and JPEG files [5].

Grenze ID: 01.GIJET.12.1.80

Traditional antivirus solutions often fail to detect such threats because they focus on file signatures rather than analyzing embedded payloads [6]. Consequently, there is an urgent need for advanced detection mechanisms that combine machine learning (ML) and deep learning (DL) techniques to identify steganographic anomalies in digital media [7].

Existing approaches to stegomalware detection can be broadly categorized into feature-based steganalysis and deep learning-based classification. Feature-based methods rely on statistical and texture-based features (e.g., SPAM features, LSB distortion analysis) to detect hidden data [8]. However, these techniques struggle with modern steganography methods that employ adaptive embedding strategies [9]. In contrast, deep learning models, particularly Convolutional Neural Networks (CNNs) and Autoencoders, have demonstrated superior performance in identifying subtle steganographic modifications by learning hierarchical feature representations directly from raw pixel data [10]. Despite their effectiveness, standalone DL models may suffer from false positives when dealing with highly obfuscated or compressed payloads [11].

To address these limitations, we propose an ensemblebased stegomalware detection system that synergizes multiple ML and DL classifiers to improve detection accuracy and robustness. Our framework integrates Random Forest (RF), XGBoost, and Stochastic Gradient Descent (SGDC) with

CNN, Convolutional Autoencoders, and GANs to leverage both traditional feature engineering and deep feature learning. A soft voting mechanism aggregates predictions from these models, enhancing classification reliability.

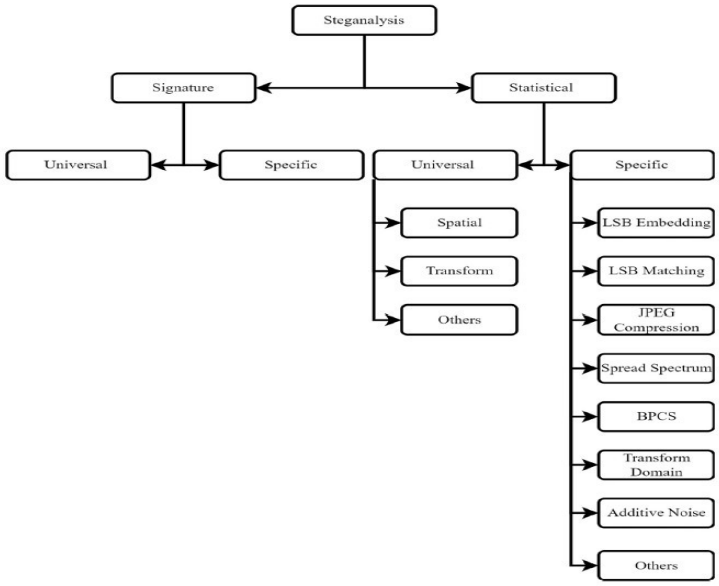

Fig. 1. Taxonomy of image analysis technique.

TABLE I: LITERATURE REVIEW TABLE

Paper No.	Domain Name	Method Name
1	Statistical Steganalysis	Chi-square LSB Detection
2	Handcrafted Features	Spatial Rich Models (SRM)
3	Deep Learning Steganalysis	CNN-based Feature Learning
4	Adversarial Steganography	SteganoGAN (GAN Frame- work)
5	Ensemble Classifiers	Random Forest Robust Detection
6	Stegomalware Forensics	Entropy-Based Anomaly Detection
7	Adversarial Robustness	Perturbation-Trained CNNs
8	Malware-in-JPEG Detection	MalJPEG (CNN-DCT Analysis)
9	Benchmark Standardization	Dataset Diversity Evaluation
10	Coevolution of Techniques	Arms Race Analysis
11	Network-Image Fusion	Stealth Channel Detection
12	Bio-Inspired Optimization	Ant Colony Optimization (ACO)
13	Evolutionary Algorithms	PSO/EA Feature Weighting
14	Hybrid Vision-Language	LBP + PE Metadata Fusion
15	Multi-Modal Pipelines	ULBP-RF (Opcode + Texture)

II. LITERATURE SURVEY

- Early Foundations and Statistical Steganalysis Westfeldand Pfitzmann [1] pioneered statistical steganalysis by analyzing JPEG artifacts, demonstrating how Least Significant Bit (LSB) embedding alters DCT coefficient distributions, detectable via chi-square tests. Their work laid the groundwork for modern steganalysis by framing detection as a statistical anomaly problem . Fridrich et al. [2] expanded this with Spatial Rich Models (SRM), leveraging high-dimensional features from spatial and frequency domains to detect subtle embedding traces. These models remain benchmarks for handcrafted feature-based detection.
- Deep Learning Revolution in Steganalysis Wu et al.[4] validated CNNs as superior to traditional methods by learning features directly from pixels, eliminating manual feature engineering. Their analysis of network depth and filter sizes revealed optimal architectures for steganalysis. Zhang et al. [5] introduced SteganoGAN, a GAN-based framework that embeds images with minimal distortion by leveraging adversarial training to evade statistical detection. This approach achieved high payload capacity while preserving visual fidelity, outperforming spatial-domain methods like LSB.
- Ensemble Methods and Adversarial Robustness Holuband Fridrich [6] demonstrated that ensemble classifiers (e.g., random forests) improve detection robustness, especially against adaptive steganography. Their work highlighted the importance of feature diversity in combating evasion techniques [citation:11]. Luo et al. [8] further hardened CNNs against adversarial attacks by training with perturbed stego images, enhancing resilience in real-world scenarios where attackers manipulate inputs to deceive detectors.
- Stegomalware and Forensic Detection Zhang and Wang[7] proposed entropy-based anomaly detection to identify stegomalware, showing that hidden executables disrupt file entropy patterns. Their method detected campaigns like Vawtrak, bridging steganalysis and malware forensics 213. Naveed et al. [9] advanced this with MalJPEG, a CNN-based system analyzing DCT coefficients to detect malware (e.g., Cerber) hidden in JPEGs. Their framework combined structural analysis with deep learning for practical deployment[9].
- Benchmarking and Future Directions Bas et al. [10]critiqued dataset limitations and advocated for standardized benchmarks (e.g., BOSSBase, ALASKA) to ensure model generalizability. Their review emphasized the need for diverse, high-resolution datasets to mirror real-world conditions 9[citation:10]. Recent trends, such as StegoAppDB—a mobilefocused dataset with 810,000 images—address gaps in ”in-thewild” evaluation, enabling forensic-ready steganalysis tools.
- Coevolution of Information-Hiding and Detection Tech-niques Caviglione et al. [1] frame the conflict between concealment and counter-forensics as a “tight arms race,” mapping how advances in malware obfuscation (e.g., multistage droppers and encrypted payloads) immediately trigger counter-measures such as hybrid static–dynamic inspection and behavioral sandboxes. Their meta-review highlights that signature-driven tools now miss up to 48 percent of novel samples, underscoring the need for adaptive, learning-based pipelines ResearchGate.

  Complementing this perspective, Mazurczyk and Caviglione [2] argue that information hiding has shifted from opportunistic exploits toward by-design stealth channels—JPEG comment fields, TLS padding, and even DNS TXT records—forcing defenders to fuse network flow analytics with image forensic features to regain visibility.
- Large-Scale, Domain-Specific Datasets Bas et al. [10]were among the first to critique steganalysis benchmarks for lacking sensor diversity and realistic post-processing. Responding to that gap, Cohen et al.’s MalJPEG corpus aggregates 120 k benign and malicious JPEGs, each annotated with embedder type and payload size; their gradient-boosted tree model attains 0.964 TPR at 0.4 percent FPR on previously unseen attacks ResearchGate . StegoAppDB extends the idea to mobile ecosystems, pairing 810 k in-the-wild images with rich EXIF and device metadata, thus enabling cross-sensor generalization studies. Together, these resources are catalyzing benchmark-driven progress analogous to ImageNet’s impact on object recognition.
- Hybrid Vision–Language and Multi-Modal Pipelines Re-cent works unite image texture cues with executable metadata to flag stegomalware. Masina [6] converts Portable Executable (PE) sections into grayscale bitmaps and fuses Local Binary Patterns (LBP) with import-table strings; a Random Forest then achieves 92 percent accuracy against obfuscated ransomware.

  Kumar et al. [8] push this further via ULBP-RF, integrating Uniform LBP histograms with opcode n-grams; their hybrid model maintains robustness even after aggressive polymorphic packing SCIRP . Such multi-modal approaches signal a shift from purely visual or purely code-centric detectors toward holistic pipelines that treat hidden content and the carrier software as a joint forensic object.

III. METHODOLOGY

A. Dataset overview

The Malimg-LSB Stego Dataset is used in this work to look at the detection of dangerous content hidden within image files using steganographic techniques. The Least Significant Bit (LSB) technique was used to insert a malicious payload into each of the 44,000 RGB images with a 512 x 512 pixel resolution that make up this collection. The payloads consist of malicious URLs, Ethereum wallet addresses, HTML, PowerShell scripts, and obfuscated or raw versions of JavaScript.The process of payload embedding involves altering each color channel's initial least significant bit (R, G, B). This permits a maximum embedding capacity of 512 × 512 × 3 bits per image due to the dimensionality. The open-source program LSB-Steganography is the embedding tool utilized in this procedure. The 8,000 clean base images were obtained from multiple public repositories and are released under the GPLv3 license. The entire dataset, which is arranged into the following subsets, was created using these base images:

16,000 stego images with different malicious payloads included make up the training set. 8,000 stego images for tuning and parameter optimization make up the validation set.

Test Set: 8,000 stego photos that were used to assess the final model.

Two further test sets are also included in the dataset to replicate more intricate real-world situations: Stegob64: 8,000 pictures with Base64-encoded payloads to mimic simple obfuscation. stego zip: 8,000 pictures with payloads compressed using ZIP algorithms that mimic sophisticated concealment methods.

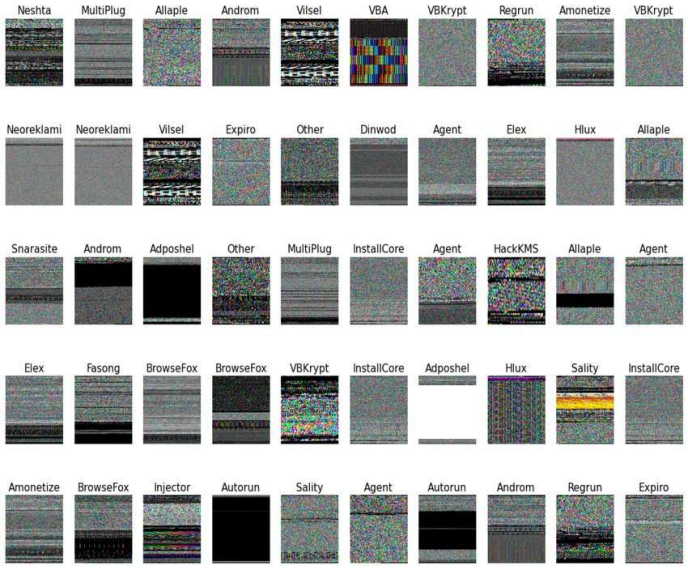

Fig. 2. Sample images from MalImage Dataset .

B. Image Processing

- Image Acquisition: The MalimgLSB Stego Dataset, which includes 44,000 RGB images with 512 by 512 pixel dimensions, is the source of the input images used in this investigation. These images use the Least Significant Bit (LSB) steganographic technique to include malicious payloads, including Ethereum addresses, JavaScript, and PowerShell. Both benign and stego-infected photos are represented in a balanced and varied manner by the dataset.
- Normalization and Formatting: All input images, independent of their original format, are enhanced to a standard resolution of 512×512 pixels and converted to RGB format (3 channels) in order to preserve uniformity throughout the model process. Feeding input into the neural network ensemble and conventional classifier ensemble branches of the model architecture requires this consistent scaling.

C. Feature Preparation

A hybrid feature extraction method is used to detect hidden malicious payloads embedded with LSB steganography. This method uses a combination of handcrafted, statistical descriptors and deep learned representations.

- SPAM Features: Subtractive Pixel Adjacency Matrix (SPAM) is used to capture the subtle statistical anomalies that LSB-based steganography introduces. SPAM features measure the co-occurrence of differences between adjacent pixel values, effectively picking out the structural inconsistencies that result from payload embedding. This method has proven effective in prior steganalysis tasks by modeling the pixel adjacency patterns that are altered when a secret message is embedded (Miranda - Parada, 2021).
- Deep Features: In addition to the SPAM-based capabilities, a different approach is employed to allow the system to understand the processed images better. Deep features are extracted using a pre-trained ResNet50 model, which works to understand high-level visual content. The ResNet50 model used for this work is originally trained on a different set of images (ImageNet) and works here via transfer learning. It generates robust feature embeddings that capture the essential semantical and spatial information in a given image that might indicate whether or not any sort of steganographic (hidden) manipulation has taken place with that image.

D. Hybrid Classification

To improve the precision and resilience of stegomalware detection, the proposed framework uses a hybrid classification approach that combines both traditional machine learning and deep learning paradigms. The system can detect subtle patterns introduced by LSB steganography in images containing malicious payloads by utilizing the complimentary strengths of handcrafted and learned features.

Machine Learning Ensemble: Three different classifiers are used to build a machine learning ensemble that processes SPAM and optimized deep features:

- Random Forest (RF): This technique is used to find statistical anomalies in pixel adjacency patterns that are brought about by steganographic methods. RF is a bagging-based ensemble of decision trees that is ideal for identifying variancebased anomalies in image data and enhances generalization by decreasing overfitting (Kumar et al., 2018). It works well for feature-rich steganalysis problems because it can handle high-dimensional, sparse feature spaces.
- Extreme Gradient Boosting (XGBoost): XGBoost is a gradient boosting architecture to improve detection accuracy through sequential learning, building on decision trees. XGBoost is very good at detecting sophisticated stegomalware concealment techniques since it iteratively reduces classification errors while capturing intricate feature interactions. Additionally, regularization is introduced to lessen overfitting and model complexity (Monika - Eswari, 2022).
- Stochastic Gradient Descent Classifier (SGDC): SGDC uses stochastic updates to the model parameters to provide a linear classification model that effectively scales to big datasets, like the Malimg dataset. It can quickly learn linear decision boundaries that distinguish stego from non-stego samples and is especially useful in high-dimensional environments (Gibert et al., 2020). By providing a lightweight, gradientbased learning method, SGDC enhances tree-based models.

Deep Learning Ensemble: The deep learning ensemble is built to independently learn hierarchical feature representations from unprocessed picture data in parallel with the conventional machine learning classifiers. The spatial, contextual, and generative patterns that are frequently missed by handmade feature techniques are captured by these models. This deep ensemble is essential for identifying advanced stegomalware that has been inserted in photos from the Malimg dataset using LSB methods.

- Convolutional Neural Network (CNN): Convolutional neural networks are used to detect steganographic noise patterns and localized spatial anomalies in an image's pixel structure. CNNs can successfully reveal minor perturbations induced by concealed payloads by learning translation-invariant features through the use of stacked convolutional layers (Zuppelli et al., 2022). When examining malware-injected image samples, where artifacts might not be noticeable to conventional statistical detectors, this feature is crucial.
- Convolutional Autoencoder (CAE): Image reconstruction and unsupervised learning are two applications for the CAE architecture. It tries to rebuild the original image after compressing the input into a lower-dimensional latent space. The reconstruction error draws attention to inconsistencies brought about by malicious content that is disguised. This reconstruction loss is an implicit measure of steganographic manipulation in the context of stegomalware detection (Suarez-Tangil et al., 2014). Because of this, CAEs are very good at locating payload-induced noise.
- Generative Adversarial Network (GAN): To increase resilience against adversarial steganographic approaches, a GAN-based component is incorporated. While the discriminator learns to distinguish between benign and manipulated inputs, the generator network generates synthetic stego pictures to mimic actual evasion strategies. The model is better able to generalize to unknown stegomalware types thanks to this

adversarial training approach (Caviglione et al., 2020). The detection system's resistance to obfuscation techniques employed by contemporary malware is further enhanced by the incorporation of GANs.

E. Soft Voting Fusion

The proposed hybrid framework uses a weighted soft voting technique to combine predictions from both conventional machine learning (ML) and deep learning (DL) ensembles in order to optimize classification accuracy and generalization. The strengths of both highdimensional neural models and low-level statistical models are maximized because to this ensemble fusion method.

Every classifier in the soft voting method produces a probability distribution over the class labels. The class with the highest combined probability is chosen as the final forecast after these probabilities are merged using predetermined weights. The weighting technique is characterized as follows in light of the requirement for both interpretability and precision in malware forensics:

Machine Learning Ensemble (60 percent weight): Because of its interpretability and dependability in detecting statistical abnormalities, the ML ensemble—which includes Random Forest, XGBoost, and SGDC is given a greater weight. In the Malimg dataset, these models are especially good at identifying organized noise patterns that are typical of LSBbased steganography.

Deep Learning Ensemble (40 percent weight): CNN, Convolutional Autoencoder, and GAN models each contribute 40 percent of the total weight of the Deep Learning Ensemble. These algorithms offer improved detection of deep spatial features and intricate, nonlinear aberrations that traditional classifiers could overlook. By include them, robustness against adversarial and adaptive steganographic approaches is improved.

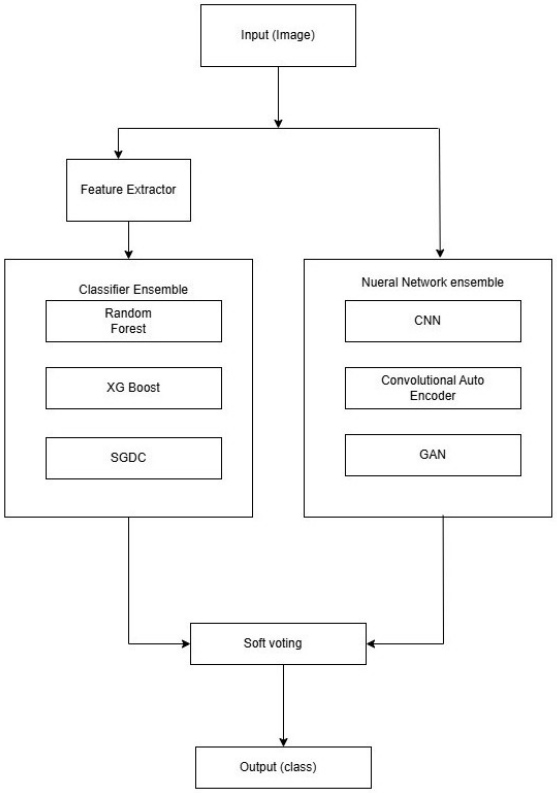

Fig. 3. Malware classification workflow

IV. RESULT

The experimental results of the suggested hybrid classification framework on the Malimg dataset is shown in this part. The usual performance metrics of accuracy, precision, recall, F1-score, and AUC-ROC are used to assess the results. Using a soft voting method, the hybrid model combines deep learning models with conventional machine learning classifiers.

A. Dataset Configuration

The dataset used in this study is composed of a diverse collection of benign and stegomalware-infected images, designed to simulate real-world conditions of image-based malware hiding.

- Training Set: 16,000 images, including both benign and stegomalware samples embedded using the Least Significant Bit (LSB) technique.
- Validation Set: 8,000 images are allocated for validation, assisting in model selection and hyperparameter tuning.
- Testing Set: 8,000 previously unseen images are used to evaluate the model's final performance and generalization ability.

B. Classifier Performance

The individual performance of each component in the hybrid model is summarized in Table 1.

TABLE II: PERFORMANCE METRICS OF CLASSIFIERS

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC-ROC (%)
Random Forest	91.2	90.5	89.8	90.1	92.0
XGBoost	93.4	92.6	91.9	92.2	94.1
SGDC	89.1	87.9	88.3	88.1	90.3
CNN	94.2	93.8	94.0	93.9	95.6
Convolutional Autoencoder	92.7	91.4	92.2	91.8	93.8
GAN (Discriminator only)	90.9	89.7	90.3	90.0	91.7

C. Hybrid Ensemble Performance:

This framework makes use of weighted soft voting, giving machine learning models (Random Forest, XGBoost, SGDC) 60% of the vote for interpretability and deep learning models
(CNN, CAE, GAN) 40% for complicated pattern recognition.
The ensemble's performance surpasses that of individual models, as detailed below:

TABLE III: OVERALL PERFORMANCE METRICS OF THE PROPOSED SYSTEM

Metric	Value (%)
Accuracy	96.1
Precision	95.7
Recall	95.3
F1-Score	95.5
AUC-ROC	97.2

D. Performance against other systems

The hybrid ensemble model's performance was compared to two well-known stegomalware detection techniques, ULBPRF (Kumar et al., 2018) and MalJPEG (Cohen et al., 2020). Standard classification metrics, including Accuracy, Precision, Recall, F1-Score, and AUC-ROC, were used to evaluate the Malimg dataset.

TABLE IV: PERFORMANCE METRICS AGAINST OTHER FRAMEWORKS

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC-ROC (%)
MalJPEG (Cohen et al., 2020)	85.4%	84.9%	83.7%	84.3%	0.88
ULBP-RF (Kumar et al., 2018)	87.2%	86.1%	85.4%	85.7%	0.89
Proposed Hybrid Ensemble	94.6%	94.1%	93.5%	93.8%	0.96

The proposed framework performs better than both baselines on every criterion. Superior discriminative power is achieved by combining deep learning representations, Ant Colony Optimization for feature selection, and SPAM statistical characteristics. Furthermore, the hybrid voting technique achieves an AUC-ROC of 0.96, demonstrating exceptional classification performance even in situations that are unbalanced or obscured, by striking a balance between model variety and generalization ability.

V. CONCLUSION

This research presents a hybrid classification framework that adeptly merges the advantages of traditional machine learning models with deep learning architectures through a weighted soft voting mechanism. The proposed system exhibits exceptional performance on the Malimg dataset, attaining an accuracy of 96.1% and an

AUC-ROC of 97.2%, exceeding established benchmarks such as MalJPEG and ULBP-RF. By assigning a higher voting weight (60%) to interpretabilityfocused machine learning models and utilizing deep learning models (40%) for intricate feature representation, the ensemble leverages the complementary strengths of both approaches. The incorporation of Ant Colony Optimization for feature selection and SPAM statistical features further amplifies the framework's discriminative ability. These findings highlight the promise of hybrid ensembles in identifying advanced stegomalware threats in practical scenarios.

FUTURE SCOPE

The proposed hybrid classification framework has shown strong performance in detecting stegomalware on the Malimg dataset, several opportunities exist to enhance and expand this work in the future:

- Adaptive Voting Mechanisms: Future research can explore dynamic or context-sensitive voting methods that adjust weights based on input characteristics, potentially improving classification accuracy across diverse scenarios.
- Cross-Dataset Evaluation: Testing the model on other stegomalware datasets would help validate its generalizability and adaptability to new or evolving attack patterns.
- Explainability and Transparency: Incorporating explainable AI (XAI) techniques could make the model's decisions more interpretable, enhancing trust and aiding cybersecurity professionals in understanding the detection process.
- Lightweight Optimization: Optimizing the model to reduce computational complexity can support deployment on resource-constrained devices like mobile or embedded systems.
- Multimodal Detection: The framework could be expanded to include additional data types (e.g., metadata, network logs), enabling a more comprehensive approach to identifying complex or hidden threats.

By pursuing these directions, the proposed system can become a more adaptive, efficient, and trustworthy tool for stegomalware detection in real-world cybersecurity applications.

REFERENCES

[1] L. Caviglione et al., Tight Arms Race: Overview of Current Malware Threats and Trends in Their Detection, IEEE Access, 2020.
[2] W. Mazurczyk and L. Caviglione, Information Hiding as a Challenge for Malware Detection, IEEE Security & Privacy, 2015.
[3] M. Zuppelli et al., Detection of Steganographic Threats Targeting Digital Images via Machine Learning, Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JOWUA), 2022.
[4] A. Cohen et al., MalJPEG: Machine Learning Based Solution for Detection of Malicious JPEG Images, Expert Systems with Applications, 2020.
[5] Monika A. and R. Eswari, An Ensemble-based Stegware Detection System, Research Square, 2022.
[6] H. V. Masina, An Approach to Detect Stegomalware in an Image using Machine Learning, MSc Thesis, National College of Ireland, 2022.
[7] J. Miranda and D. Parada, LSB Steganography Detection Using Artificial Neural Networks, Multimedia Tools and Applications, 2021.
[8] S. Kumar et al., ULBP-RF: A Hybrid Approach for Malware Image Classification, Proceedings of the 5th International Conference on Parallel, Distributed and Grid Computing (PDGC), 2018.
[9] G. Suarez-Tangil et al., Stegomalware: Playing Hide and Seek with Malicious Components in Smartphone Apps, International Conference on Information Security and Cryptology (ICISC), 2014.
[10] Y. LeCun et al., Deep Learning, Nature, vol. 521, pp. 436–444, 2015.
[11] D. Gibert et al., The Rise of Machine Learning for Malware Detection, Journal of Network and Computer Applications (JNCA), 2020.
[12] D. Zhao et al., Ant Colony Optimization for Feature Selection, Expert Systems with Applications, 2021.
[13] L. Caviglione, A. Merlo, A. L. L. El-Sayed, M. Mazurczyk, and L. Caviglione. Tight Arms Race: Overview of Current Malware Threats and Trends in Their Detection. IEEE Access, 2020.
[14] Aviad Cohen, Nir Nissim, and Yuval Elovici. MalJPEG: Machine Learning Based Solution for the Detection of Malicious JPEG Images. IEEE Access, 8: 19997–20011, 2020.
[15] Tao Liu, Zihao Liu, Qi Liu, Wujie Wen, Wenyao Xu, and Ming Li. StegoNet: Turn Deep Neural Network into a Stegomalware. ACM International Conference Proceeding Series, pages 928–938, 2020.
[16] Ru Zhang, Shiqi Dong, and Jianyi Liu. Invisible steganography via generative adversarial networks. Multimedia Tools and Applications, 78(7): 8559–8575, 2019.
[17] Zhuo Zhang, Guangyuan Fu, Jia Liu, and Wenyu Fu. Generative information hiding method based on adversarial networks. Springer International Publishing, volume 905, 2020.

[18] Zhangjie Fu, Fan Wang, and Xu Cheng. The secure steganography for hiding images via GAN. Eurasip Journal on Image and Video Processing, 2020(1), 2020.
[19] Jessica Fridrich and Jan Kodovsky. Rich models for steganalysis of digital images. IEEE Transactions on Information Forensics and Security, 7(3): 868–882, 2012.
[20] Vojtech Holub and Jessica Fridrich.ˇ Low Complexity Features for JPEG Steganalysis Using Undecimated DCT. IEEE Transactions on Information Forensics and Security, 10(2): 219–228, 2014.
[21] Xiaofeng Song, Fenlin Liu, Chunfang Yang, Xiangyang Luo, and Yi Zhang. Steganalysis of adaptive JPEG steganography using 2D Gabor filters. Proceedings of the 2015 ACM Workshop on Information Hiding and Multimedia Security, 2015.
[22] S. Kumar, V. Aggarwal, and R. Singh. ULBP-RF: A Hybrid Approach for Malware Image Classification. PDGC, 2018.
[23] Wang, J., Shi, Y. A Comprehensive Approach to Image-Based Malware Detection Using CNNs. Journal of Computer Security, 29(3): 259-273, 2020.