https://doi.org/10.33472/AFJBS.6.11.2024.1268-1278

African Journal of Biological Sciences

Journal homepage: http://www.afjbs.com

Research Paper Open Access

ADVANCING STEGANALYSIS: EXPLORING FEATURE EXTRACTION TECHNIQUES AND CLASSIFIERS FOR ENHANCED DETECTION

Gargi Kalia[1], Kamal Malik[2]

[1]CT University (Ludhiana), India.
[2]Professor, CSE, CT University (Ludhiana), India.

Email: [1]Gargikalia1@gmail.com, [2]kamal17203@ctuniversity.in

Article Info

Volume 6, Issue 11, July 2024

Received: 23 May 2024

Accepted: 20 June 2024

Published: 09 July 2024

doi: 10.33472/AFJBS.6.11.2024.1268-1278

ABSTRACT:

Steganalysis, the technique of identifying hidden information in digital media, is becoming increasingly important in security and forensic applications. This article describes a way for expanding steganalysis approaches by incorporating optimized feature extraction techniques to increase detection rates. This article explains various feature extraction techniques and classifiers to improve the distinction between stego and cover media. It focus on the overview of the current state of steganalysis and highlight the limitations of existing techniques for better detection. This study contributes to the ongoing growth of steganalysis approaches by identifying promising pathways for improving security measures and forensic investigations in digital environments. In field of biological sciences, where they can help safeguard sensitive biological data from cyberattacks. Ensuring the security of this data is crucial for the advancement of biological sciences, as well as for preserving the integrity of scientific research and protecting privacy.

Keywords: Image steganalysis, Stego image, Hidden data, Classifier, Feature extraction, Detection rates, Digital media, Digital forensics

1. INTRODUCTION

The widespread use of digital media in modern communication and information exchange has resulted in an increase in strategies for concealing crucial information within seemingly innocuous content. Steganography, or the method of concealing messages or data within other data, poses substantial issues for digital security and forensic investigations. Traditional steganalysis procedures frequently rely on manual examination or basic statistical analysis, which may be insufficient to discover complex steganographic processes. In order to improve openness, accuracy, and security of its services for the public, the federal government and some state governments have taken the initiative to successfully implement e-government in a number of service areas using information and communication technology[24]. The current commercial systems are primarily designed for use by organisations and governments that have strict security needs[25].

Furthermore, the rising use of high-capacity embedding techniques poses a significant challenge to traditional detection methods. As a result, there is an urgent need for better steganalysis approaches capable of obtaining improved detection rates in the face of evolving steganographic techniques. Feature extraction is critical in steganalysis because it allows us to identify discriminative trends between cover and stego media[1]. We will present an overview of the current state of steganalysis and highlight the limitations of existing techniques. We then describe the goals of this study, which include establishing an enhanced steganalysis methodology and assessing its usefulness in increasing detection rates. [2].

2. Steganalysis Methods

Steganalysis approaches are often classified into two types: signature steganalysis and statistical steganalysis, according to the method employed to uncover hidden signals in digital media. Signature steganalysis focuses on identifying specific signatures or patterns associated with the presence of embedded information, while statistical steganalysis relies on detecting changes in statistical properties induced by the embedding process.

Within signature steganalysis, there are further classifications, including specific signature steganalysis and universal signature steganalysis. Specific signature steganalysis aims to identify unique signatures associated with particular embedding algorithms or techniques, whereas universal signature steganalysis seeks to detect common signatures shared across various embedding methods.

Statistical steganalysis, on the other hand, leverages statistical properties altered by the embedding process to detect hidden information. This approach is considered powerful due to its sensitivity and encompasses various methods. Examples of steganalysis include LSB embedding, LSB matching, spread spectrum, JPEG compression, additive noise, and transform domain [3].

Recent advancements in steganalysis methods have introduced techniques such as block-based steganalysis and the Spatio-Color Rich Model. The block-based steganalysis method operates through a training process and a testing process, where images are decomposed into blocks, features are extracted, A tree-structured clustering technique is used to classify blocks according to their features. In testing

phase, blocks are classified, and a majority voting rule is used for decision-making. The Spatio-Color Rich Model utilizes both spatial and color features for steganalysis, with components such as SRMQ1 and CRMQ1 calculated to detect steganography efficiently. The model employs collaborative representation to solve least square problems, effectively utilizing training samples from both cover and stego images[4].

LSB matching steganalysis focuses on detecting differences in neighboring pixels before and after data embedding, utilizing patterns created from significant information of neighboring pixels. The Difference Histogram Characteristics Function (DHCF) achieves accurate steganography detection in color images by analyzing differences in pixel gray values and extracting features from non-adjacent pixels.

These advancements represent significant progress in steganalysis methodologies, offering improved accuracy and efficiency in detecting hidden information within digital media[5].

3. Image Processing Techniques

Image preprocessing involves employing techniques and methods to alter or improve digital images before they undergo further analysis or processing by either algorithms or humans. The fundamental goal of image preprocessing is to improve the quality, clarity, or interpretability of images, making them more appropriate for specific activities or applications.

Typically, common image preprocessing techniques encompass:

1. **Noise Reduction:** Eliminating or minimizing noise, such as random variations in pixel values, which may arise from sensor limitations, transmission interference, or environmental factors.
2. **Contrast Enhancement**: Adjusting the contrast of an image to enhance the visibility of features or details by augmenting the disparity between light and dark areas.
3. **Normalization**: Ensuring uniform brightness and color levels across images to mitigate disparities resulting from variations in lighting conditions or camera configurations. Adjusting pixel values to a standardized range, often between 0 and 1, to facilitate processing and comparison between images.
4. **Resizing and Scaling:** Modifying the dimensions of an image to a desired size or resolution, which can be advantageous for standardizing inputs or reducing computational complexity.
5. **Image Registration:** Aligning multiple images of the same scene to rectify variations in perspective, rotation, or scale.
6. **Color Correction:** Modifying color balance, saturation, or hue to rectify inaccuracies or discrepancies introduced during image acquisition.
7. **Edge Detection:** Identifying and enhancing the edges or boundaries of objects within an image, which aids in segmentation or feature extraction.
8. **Filtering:** Applying diverse filters to either smooth or sharpen images, eliminate undesired artifacts, or accentuate specific features.
9. **Image Fusion:** Integrating multiple images of the same scene captured from distinct sensors or modalities to generate a composite image enriched with additional information[6].

4. Feature Extraction Techniques

Feature extraction techniques are methods of obtaining useful information or characteristics (features) from raw data. In image processing or computer vision, feature extraction techniques strive to find fundamental patterns or structures in images, which may subsequently be used for a range of applications

such as classification, object identification, and recognition. Some common feature extraction techniques in image processing are:

4.1. Local Binary Patterns (LBP)

LBP represent a technique employed in image processing and computer vision for feature extraction. It functions by evaluating the intensity of a central pixel against its neighboring pixels, thereby generating binary patterns that encode local texture details. These patterns are utilized to construct LBP histograms or texture maps, serving as features for tasks like texture classification, facial recognition, and object detection[7,8].

4.2. Spatial Frequency Texture Analysis (SFTA)

SFTA is a technique used in image processing and computer vision to analyse textures using their spatial frequency content. It involves decomposing an image into different frequency bands using techniques such as Fourier or wavelet transforms. By examining the distribution of spatial frequencies across these bands, SFTA can extract valuable information about the texture properties of the image. This methodology is frequently utilised in applications such as texture classification, segmentation, and image retrieval[7].

4.3. Gray-Level Co-occurrence Matrix (GLCM)

GLCM is an important feature extraction method for image processing and texture analysis. It quantifies the frequency of pairings of pixel intensity values at predetermined distances and directions and stores this information in a matrix. From the GLCM, statistical measures such as contrast, correlation, energy, and entropy are derived, offering insights into the texture properties of the image. Widely utilized in texture classification, segmentation, and image retrieval, GLCM features enable the discernment of distinct textures or regions within an image based on their spatial arrangements of pixel intensities [7,8].

4.4. Discrete Cosine Transform (DCT)

The differential Discrete Cosine Transform (DCT) coefficients are used to calculate the absolute values of Neighbouring Joint probability density (absNJ) features. To distinguish between stego and cover pictures, apply the $PPtth$ power of DCT coefficients, rather than their first power gains. Both inter-block and intra-block aspects of DCT coefficients are extracted, yielding a variety of features. Intra-block features capture the frequency dependence of DCT coefficients, whereas inter-block features remove spatial relationships between DCT blocks. The most difficult coefficient relationships are retrieved by extending the original features to extended versions, resulting in final features [9].

4.5. Spectrum-Based Feature Extraction

Spectrum-based feature extraction employs a mix of DCT and Discrete Fourier Transform. The image is transformed into the frequency domain to remove redundant information, and a subset of altered coefficients is chosen to preserve critical features for recognition. DFT is applied to the pre-processed image, with low-frequency components in the middle of the spectrum. The centre piece is removed with a centred rectangular mask, and DCT is applied to the DFT spectrum to improve recognition rate. The DWT-Dual Subband Frequency Domain Feature Extraction (DDFFE) method combines DWT, DFT, and DCT to efficiently extract, translate, and illuminate invariant features. DWT uses approximation coefficients in addition to horizontal coefficients in 2-dimensional images, whereas DFT compensates for DWT's translation variance difficulties by extracting the image's frequency characteristics[10]. A quadruple ellipse mask is used to isolate low-frequency components from the centre of the DFT spectrum. DFT includes magnitude and phase information, which are retrieved individually, and it is applied to the preprocessed image, shifting low-frequency components from the image's corner to the centre.

4.6. Non-Sampled Contourlet Transform (NSCT)

The Logarithmic Non-Sampled Contourlet Transform (NSCT) retrieves image invariant information such as strong and weak edges, as well as noise. The extraction and recombination of these features yields two types of components: illumination and reflectance. The illumination component includes the image's low-pass subband as well as the low-frequency portion of strong edges, whereas the reflectance component includes pathetic edges and high-frequency portions of strong edges. When the facial image is influenced by noise, the reflectance component is used to remove such noise. The NSCT algorithm successfully captures contours in the original image [11,12].

5. Machine Learning Based Classification Techniques

The steganalyst improves the classifier's training by combining increasingly complex cover models and larger datasets, resulting in more precise and resilient detectors.

5.1 Ensemble Classifier

This classifier creates a steganography detector by choosing a model for the cover source and training it on a collection of cover and stego photos. It combines the decisions of separate base learners to reach a final decision. To construct a robust detection system, the ensemble classifier combines the strengths of multiple base learners, including decision trees, support vector machines, and neural networks. By merging the outputs of numerous models, it can successfully capture diverse patterns and characteristics present in both cover and stego images, increasing the total detection accuracy [13].The ensemble classifier offers a rapid method for building a steganography detector without extensive manual tuning. It automates the process of selecting and training base learners, streamlining the development pipeline. By integrating the decisions of multiple base learners, the ensemble classifier achieves enhanced accuracy compared to individual classifiers. It reduces the risk of overfitting and generalization errors by aggregating diverse perspectives on the detection problem. The classifier combines the outputs of individual base learners using techniques such as voting or averaging to arrive at a final decision. This aggregation process mitigates the impact of noise and uncertainty in the data, leading to more robust detection outcomes.

5.2. Ensemble Based Extreme Learning Machine (EN-ELM) Classifier

EN-ELM incorporates ensemble learning and cross-validation during the training phase. It creates an ensemble of predictors using various random parameters and makes decisions via a majority voting mechanism. This classifier is particularly adept at dealing with high-dimensional data and is noted for its efficiency and robustness in classification tasks [14]. EN-ELM uses ensemble learning techniques to merge numerous base learners, each trained on a distinct sample of the data or with different parameter settings. Cross-validation is used to guarantee that the ensemble generalises effectively to new data, hence improving its reliability and performance. EN-ELM stands as a powerful classifier that combines the benefits of ensemble learning, cross-validation, and extreme learning machines, offering a reliable and efficient solution for classification tasks in diverse domains. Its ability to harness the collective intelligence of multiple predictors while maintaining computational efficiency makes it a valuable tool in the arsenal of machine learning practitioners. EN-ELM stands as a powerful classifier that combines the benefits of ensemble learning, cross-validation, and extreme learning machines, offering a reliable and efficient solution for classification tasks in diverse domains. Its ability to harness the collective intelligence of multiple predictors while maintaining computational efficiency makes it a valuable tool in the arsenal of machine learning practitioners [14, 15].

5.3. Extreme Learning Machine (ELM) Classifier

The Extreme Learning Machine (ELM) classifier is extremely effective for multi-label classification tasks with big datasets, particularly when using single hidden layer feedforward neural networks. Unlike standard neural network models, which need precise adjustment of hidden node parameters, ELM takes a novel approach by randomly generating these values, resulting in a fast and efficient training procedure [16,17].

5.4. Cognitive Ensemble ELM Classifier

The Cognitive Ensemble ELM Classifier is a sophisticated classification model that uses ensemble learning principles and cognitive mechanisms to improve classification accuracy. Based on the hinge loss function, this classifier creates separate Extreme Learning Machine (ELM) classifiers, each trained to reduce classification mistakes. One distinctive element of this classifier is its cognition computation, which entails taking the weighted sum of separate

classifier outputs. This weighted total represents the confidence or certainty in each classifier's conclusion. By aggregating the outputs in this manner, the classifier benefits from the collective intelligence of the ensemble and assigns higher priority to more reliable classifiers.
Moreover, the Cognitive Ensemble ELM Classifier dynamically enhances the winning classifier based on correctly classified samples. This adaptive mechanism allows the classifier to continuously improve its performance over time by reinforcing successful decision-making strategies. The winning classifier, i.e., the classifier with the highest contribution to the ensemble's decision, is dynamically enhanced based on correctly classified samples. This reinforcement mechanism allows the classifier to adapt and improve its decision-making strategy based on real-world feedback [18].

5.5. Support Vector Machine (SVM) Classifier

The Support Vector Machine (SVM) Classifier is a popular machine learning approach for binary classification applications. It seeks to identify the best linear decision surface, known as the maximum margin hyperplane, for effectively separating different class feature vectors in the input space [19].
SVM seeks the hyperplane that maximises the margin between classes in the feature space. This margin sets the distance between the hyperplane and the nearest data points, allowing for accurate generalisation of previously unknown data. LIBSVM and other SVM implementations allow a variety of kernel functions, such as linear, polynomial, radial basis function (RBF), and sigmoid. These kernel functions enable SVM to capture nonlinear relationships between input features and improve classification [20].

5.6. Naive Bayes Classifier:

The Naive Bayes classifier is a basic yet powerful probabilistic model for classification problems that is widely used in text categorization, spam detection, and other applications. It uses Bayes' theorem with the naive assumption of feature independence to ease computation. It assigns the instance the most likely class label based on the probability of each class given the input attributes. Despite their simplicity and unrealistic assumption of feature independence, Naive Bayes classifiers frequently perform exceptionally well, particularly on high-dimensional datasets, thanks to their computational efficiency and capacity to manage massive amounts of data. However, they may struggle to capture complicated connections between features [21]. The mathematical representation is as follows:
The "naive" assumption in Naive Bayes is that when given a class label, features are conditionally independent. This simplifies the likelihood term computation by multiplying the probabilities of each features based on their class label. During the training phase, the method determines the prior probabilities $P(Y)$ and likelihoods $P(X|Y)$ based on the training dataset. It computes the probability for each class label and feature. The technique uses Bayes' theorem during the classification process. Naive Bayes is computationally economical and performs well with huge datasets and multidimensional feature spaces. In essence, Naive Bayes is a simple but strong classifier that works well in applications using discrete or continuous features, such as text categorization and spam filtering. Despite its simplicity and "naive" assumption, it regularly performs brilliantly in practice, particularly on datasets that reasonably meet the independence assumption [21, 22].

5.7. Gaussian Discriminant Analysis (GDA) Classifier: GDA is a statistical method for classification in machine learning. It presupposes that the properties of each class follow a Gaussian (normal) distribution. This indicates that the data points for each class form a bell-shaped curve. GDA is a generative model, which means it depicts the probability distributions of the features in each class. It obtains the parameters (mean and covariance) of these Gaussian distributions from the training dataset [23].

6. Proposed Method

A flowchart of the suggested method is included in this section. Algorithm was developed as explained below.

6.1 Algorithm for Image Classification using Feature Extraction and Multiple Classifiers

Algorithm: Proposed Steganalysis Classification_based AlexSMA AND T-AlexSMA
INPUT: Image Dataset.
OUTPUT: Non-stego/Stego Image.
Begin
For
1. Read each image by "Imread ( )" function
2. Transforming the RGB image to the gray image using "rgb2gray ( )" function;
3. For each transformed image, extract the SFTA features {Sftaf1, Sftaf2. . . Sftaf21} to obtain a 21-dimension feature vector
4. For each transformed image, extract the LBP features {Lbpf1, Lbpf2. . . Lbpf59} to obtain a 59-dimension feature vector
5. Extract the GLCM features vector,create the co-occurrence matrix for each transformed image, using "graycomatrix ( )" function. Also, Extract the GLCM features {Glcmf1, Glcmf2, Glcmf3, Glcmf4 . . . Glcmf14} to obtain a 14-dimension feature vector.
6. Train the AlexSMA classifier with these feature vectors and T-AlexSMA classifier with these feature vectors.
7. Test the trained AlexSMA model to determine if the image is non-stego or stego. Test the trained T AlexSMA model to determine if the image is non-stego or stego. End for End

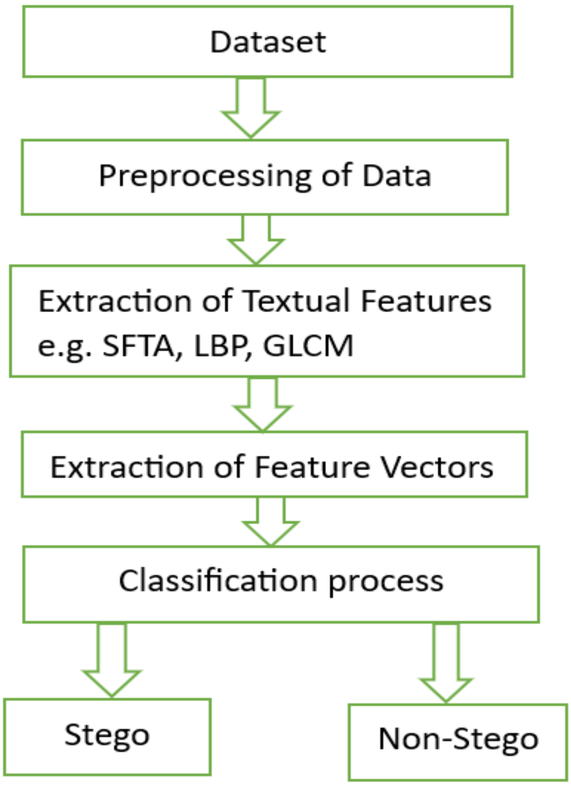

Fig. 1 Flowchart of Proposed Method

2. RESULTS

The following section elucidates the effectiveness of the proposed approach. The experiment was conducted using MATLAB 2020, employing a dataset sourced from publicly accessible websites. Specifically, the IStego100K dataset, renowned for its extensive collection of images for steganalysis, was utilized for training and testing.
The IStego100K dataset comprises 208,104 photos, with 200,000 images allocated for training purposes, consisting of 100,000 cover-stego image pairs. The remaining 8,104 images constitute the testing set. Each image in the dataset has a resolution of 1024 x 1024 pixels.

Algorithm	Accuracy
'LBP-GDA'	75
'GLCM-GDA'	79
'SFTA-GDA'	90
'LBP+GLCM+SFTA-GDA'	85
'LBP-NB'	71
'GLCM-NB'	73
'SFTA-NB'	75
'Ada-Boost'	93
'Ensemble Classifier'	83
'Naive Bayes'	60
'Ensemble Classifier'	94
'GDA'	90
'AlexSMA'	96.189
'T-AlexSMA'	98.792

Fig. 2 Performance Comparison Results of classification methods

1. **Local Pattern Analysis:**

- LBP-GDA: 75% accuracy
- LBP-NB: 71% accuracy

Conclusion: LBP-based algorithms show moderate accuracy, with LBP-NB slightly behind LBP-GDA.

Texture Information Extraction:

- GLCM-GDA: 79% accuracy
- GLCM-NB: 73% accuracy

Conclusion: GLCM-based algorithms outperform LBP-based ones, with GLCM-GDA showing the highest accuracy.

Spatial Frequency Features:

- SFTA-GDA: 75% accuracy
- SFTA-NB: 75% accuracy

Conclusion: SFTA-based algorithms perform comparably to LBP-based ones, with SFTA-GDA showing equal accuracy to LBP-GDA.

Combined Feature Sets:

- LBP-GLCM-SFTA-GDA: 85% accuracy
- LBP+GLCM+SFTA-NB: 75% accuracy

Conclusion: Combining feature sets improves accuracy significantly, with LBP-GLCM-SFTA-GDA achieving the highest accuracy among non-deep learning methods.

Deep Learning Models:

- Alex SMA (AlexNet): 96.189% accuracy
- T-Alex SMA (Transfer Learning with AlexNet): 98.792% accuracy

Conclusion: Deep learning models vastly outperform traditional methods, with T-Alex SMA showing exceptional accuracy due to transfer learning.

3. Ensemble Methods:

- Ada Boost: 93% accuracy
- Ensemble Classifier: 83% accuracy
- Unspecified Ensemble Classifier: 94% accuracy

Conclusion: Ensemble methods perform well, with Ada Boost and the unspecified Ensemble Classifier showing high accuracy.

Other Models:

- Naive Bayes: 60% accuracy
- GDA: 90% accuracy

Conclusion: Naive Bayes demonstrates lower accuracy compared to other methods, while GDA performs exceptionally well.

Future Scope

In the future, our findings underscore the importance of continued study in this rapidly growing field. Subsequent research should consider optimising algorithms for domain-specific tasks, exploring the possibility of collaborative efforts, investigating cross-disciplinary applications, and confronting moral quandaries associated to algorithmic judgement. This comprehensive method ensures the continuing progress and appropriateness of sorting algorithms for Steganalysis. One key area of focus involves evaluating the scalability of sorting algorithms, particularly concerning their ability to manage the mounting volume and intricacy of datasets encountered in real-world applications. As digital content continues to proliferate across various platforms, ranging from images to text and multimedia files, sorting algorithms must demonstrate robustness and efficiency in processing large and diverse datasets to maintain the effectiveness of Steganalysis techniques.

2. CONCLUSION

This paper gives a thorough examination of steganalysis approaches designed to improve the detection of hidden information inside cover media. The steganalyst uses various classifiers such as Ensemble classifiers, Extreme Learning Machines (ELM), Support Vector Machines (SVM), and Bayesian estimation approaches to build robust detectors that can successfully recognise steganographic information. Classifiers gain in accuracy and durability as training techniques are iteratively refined, including the addition of complicated cover models and big datasets. Significantly different accuracy rates (with T-AlexSMA being the most accurate) and competitive results from Ensemble Classifiers and Naive Bayes underscore the subtle benefits and challenges of each technique. These findings have direct implications for decision-makers and provide precise guidelines for selecting sorting algorithms that are appropriate for specific task requirements.. Looking ahead, the prospective scope of steganalysis research offers exciting opportunities for inquiry. One such approach is to develop hybrid models that combine the strengths of many classifiers to improve detection accuracy and robustness. Deep learning techniques such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) provide exciting potential for improving feature extraction and pattern recognition in steganalysis. The approach to image classification, which includes feature extraction techniques and multiple classifiers, serves as a comprehensive framework for this study. We describe the step-by-step process to address this difficulty by combining feature extraction methods and numerous classifiers which can help new researchers to work in this field.

3. REFERENCES

1. Farooq, N., & Selwal, A. (2023). Image steganalysis using deep learning: a systematic review and open research challenges. *Journal of Ambient Intelligence and Humanized Computing*, *14*(6), 7761-7793.
2. Priscilla, C. V., & HemaMalini, V. (2023). Steganalysis Techniques: A Systematic Review. *Journal of Survey in Fisheries Sciences*, *10*(2S), 244-263.
3. Farooq, N., & Selwal, A. (2023). Image steganalysis using deep learning: a systematic review and open research challenges. *Journal of Ambient Intelligence and Humanized Computing*, *14*(6), 7761-7793.
4. Lwowski, J., Corley, I., & Hoffman, J. (2023). Neural Steganalysis with Spatial Rich Models for Image Steganography Detection. *Authorea Preprints*.
5. Sharma, V. K., Mir, R. N., & Rout, R. K. (2023). Towards secured image steganography based on content-adaptive adversarial perturbation. *Computers and Electrical Engineering*, *105*, 108484.
6. Boopathi, S., Pandey, B. K., & Pandey, D. (2023). Advances in artificial intelligence for image processing: techniques, applications, and optimization. In *Handbook of research on thrust technologies' effect on image processing* (pp. 73-95). IGI Global.
7. Dalal, M., & Juneja, M. (2023). Steganalysis of DWT Based Steganography Technique for SD and HD Videos. *Wireless Personal Communications*, *128*(4), 2441-2452.
8. Prakash, K., & Saradha, S. (2023). Efficient prediction and classification for cirrhosis disease using LBP, GLCM and SVM from MRI images. *Materials Today: Proceedings*, *81*, 383-388.
9. Scribano, C., Franchini, G., Prato, M., & Bertogna, M. (2023). Dct-former: efficient self-attention with discrete cosine transform. *Journal of Scientific Computing*, *94*(3), 67.
10. Li, X., Zhang, Y., Awais, M., Naqvi, S. M. Z. A., Li, L., Chen, H., & Hu, J. (2023). Exploration of efficient SERS features extraction algorithm for rapid detection of thiabendazole residues in apples. *Lwt*, *187*, 115310.
11. Zeng, Y., Yi, W., Deng, J., Chen, W., Xu, S., & Huang, S. (2020). Remote sensing image fusion using improved IHS and non-subsampled Contourlet transform. In *Urban Intelligence and Applications: Proceedings of ICUIA 2019* (pp. 55-67). Springer International Publishing.
12. Zhang, P., Wang, L., & Eslami, H. (2022). A mathematical model of PCNN for image fusion with non-sampled contourlet transform. *Applied Mathematics and Nonlinear Sciences*, *8*(1), 2243-2252.
13. Kumar, M., Bajaj, K., Sharma, B., & Narang, S. (2022). A Comparative Performance Assessment of Optimized Multilevel Ensemble Learning Model with Existing Classifier Models. *Big Data*, *10*(5), 371-387.
14. Ye, X., He, Y., & Huang, J. Z. (2020). Attribute bagging-based extreme learning machine. In *Algorithms and Architectures for Parallel Processing: 20th International Conference, ICA3PP 2020, New York City, NY, USA, October 2–4, 2020, Proceedings, Part II 20* (pp. 509-522). Springer International Publishing.
15. Wang, J., Lu, S., Wang, S. H., & Zhang, Y. D. (2022). A review on extreme learning machine. *Multimedia Tools and Applications*, *81*(29), 41611-41660.
16. Wang, J., Lu, S., Wang, S. H., & Zhang, Y. D. (2022). A review on extreme learning machine. *Multimedia Tools and Applications*, *81*(29), 41611-41660.

17. Zhang, L., Wen, D., Li, C., & Zhu, R. (2020). Ensemble classifier based on optimized extreme learning machine for motor imagery classification. *Journal of neural engineering*, *17*(2), 026004.
18. Sachnev, V., Sundararajan, N., & Suresh, S. (2023). A New Approach for JPEG Steganalysis with a Cognitive Evolving Ensembler and Robust Feature Selection. *Cognitive Computation*, *15*(2), 751-764.
19. Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, *408*, 189-215.
20. Kurani, A., Doshi, P., Vakharia, A., & Shah, M. (2023). A comprehensive comparative study of artificial neural network (ANN) and support vector machines (SVM) on stock forecasting. *Annals of Data Science*, *10*(1), 183-208.
21. Chen, S., Webb, G. I., Liu, L., & Ma, X. (2020). A novel selective naïve Bayes algorithm. *Knowledge-Based Systems*, *192*, 105361.
22. Narayan, Y. (2021). Comparative analysis of SVM and Naive Bayes classifier for the SEMG signal classification. *Materials Today: Proceedings*, *37*, 3241-3245.
23. [23] Dan, C., Wei, Y., & Ravikumar, P. (2020, November). Sharp statistical guaratees for adversarially robust gaussian classification. In *International Conference on Machine Learning* (pp. 2345-2355). PMLR.
24. S Sirsendu , K Sunil, "Applications of Public Key Watermarking for Authentication of Job-Card in MGNREGA", BIJIT – BVICAM's International Journal of Information Techno
25. S Dilbag, S Ajit , " An Effective Technique for Data Security in Modern Cryptosystem", BIJIT – BVICAM's International Journal of Information Technology, Vol 2 No 1, pp. 189- 194, January – June 2010