Image Steganalysis using Feature Selection based on Mutual Information and Adaptive Particle Swarm Optimization

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering

in

Computer Science and Engineering

Submitted by

Jasmanpreet Kaur

(Roll no: 801632015)

Under the supervision of

Dr. Singara Singh

Associate Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY
PATIALA 147004

July 2018

Certificate

I hereby certify that the work, which is being presented in the thesis, entitled "*Image Steganalysis using Feature Selection based on Mutual Information and Adaptive Particle Swarm Optimization*", in partial fulfillment of the requirements for the award of the degree of Master of Engineering in *Computer Science and Engineering* and submitted in Computer Science Department of Thapar Institute of Engineering and Technology, Patiala is an authentic record of my own work carried out under the supervision of *Dr. Singara Singh* and refers other researcher's work which are duly listed in the reference section.

The matter presented in this thesis has not been submitted elsewhere for the award of any other degree of this or any other University.

Jasmanpreet Kaur

Jasmanpreet Kaur

This is to certify that the above statement made by the candidate is correct and true to the best of our knowledge.

Singara 15/7/18

Dr. Singara Singh

Associate Professor,

CSED

i

Abstract

In recent years, Steganalysis has been an area of active research. Feature Selection is a necessary phase of steganalysis in order to achieve high detection accuracy. Steganalytic feature selection methods based on Minimal Information *(MI)* and Adaptive Weight based Particle Swarm Optimization *(APSO)* are proposed in this work in order to effectively reduce the high space dimensionality of the statistical features used in state-of-the-art steganalysis. First the differentiability of each feature dimension is calculated using MI parameter and further weight based sorting is carried out for the whole feature dimensions. The features sorted in descending order of differentiability among another features. So first few feature dimensions are selected in this phase. In order to further optimize the feature reduction and to reduce inseparable features, weight adaptive *PSO* in used which further reduces the feature vector space to a threshold value decided. The fitness function used to get the best particles in the *PSO* is *AUC* measure of a classifier which is calculated for each particle and the particle having high AUC is chosen as final optimal selected features. The effectiveness of the proposed feature selection is carried out by classifying the combined dataset of stego and cover images taken from *BOSS*base dataset using three different machine learning techniques named as decision tress, *kNN* and *SVM*. The statistical features used for evaluation are *SPAM* and *CC-PEV* feature extraction methods which have 686 and 548 feature space dimensionality. The final feature reduction is tested at 80 and 100 selected features. Experimental results shows that high classification accuracy is achieved with the proposed features reduction methods.

Acknowledgement

First, I would like to extend my deep gratitude to my supervisor **Dr. Singara Singh** for constant supervision, for their advice and patiently guidance at every step of my ME program. Without unfailing support and perception in me, this thesis would now not have been viable. Their contribution to this thesis goes well beyond their role as an academic supervisor and includes constant support on a personal level without which this journey may never have been completed. And for this, I'm truly thankful. He is a great mentor for my life as well.

I would like to express my gratitude to **Dr. Maninder Singh**, Head of Computer Science and Engineering Department and **Dr. Ashutosh Mishra**, P.G. coordinator their constant motivation and encouragement.

I also wish to thank my research committee members and non-teaching staff of the Computer Science and Engineering Department for their help and support. I would also like to thanks to my teachers and friends from whom I learn the art of happiness and never give up approach.

Finally, I would like to express my sincere and deep gratitude to my parents and family members for their love, encouragement, care, and support.

Jasmanpreet Kaur

Table of Contents

Abstract . . . . . **ii**

Table of Contents . . . . . **iv**

List of Figures . . . . . **vi**

List of Tables . . . . . **viii**

List of Abbreviations . . . . . **ix**

Chapter 1 Introduction **1**

1.1 Steganography 1

1.2 Steganalysis 2

1.3 Approaches of Steganalysis 3

1.3.1 Static Steganalysis 3

1.3.2 Dynamic Steganalysis 3

1.4 Classification of Steganalysis 4

1.5 Universal Steganalysis 4

1.6 Features Extraction 5

1.7 Feature Selection 6

Chapter 2 Literature survey **8**

2.1 Feature Extraction Techniques: 8

2.2 Feature Selection Techniques: 12

Chapter 3 Problem Statement **17**

3.1 Research Gap 17

3.2 Statement 17

3.3 Contribution 18

Chapter 4 Proposed Technique **19**

4.1 System Module 19

4.2 Feature Extraction . . . 19
4.2.1 Spatial-domain based feature selection for steganlysis . . . 19
4.2.2 Frequency-domain based feature selection for steganlysis . . 21
4.3 Statistical Dependency and Mutual information between features and labels . . . 21
4.4 Particle Swarm Optimization . . . 22
4.4.1 Parameters Selection and diversity measure in *PSO* . . . 23
4.4.2 Adaptive inertia weight-based *PSO* . . . 26
4.4.3 Fitness Function . . . 29
4.5 Training Step . . . 30

Chapter 5 Experimental Results . . . 33
5.1 Overview . . . 33
5.2 Confusion matrix for performance evaluation . . . 33
5.2.1 Classification Accuracy . . . 35
5.2.2 Sensitivity . . . 36
5.2.3 Specificity . . . 36
5.3 Results . . . 36

Chapter 6 Conclusion and Future Work . . . 52
6.1 Conclusion . . . 52
6.2 Future Scope . . . 53

References . . . 54

List of Publications . . . 59

List of Figures

Figure No.	Title	Page No.
1.1	Block diagram of Steganography	2
1.2	Block diagram of Steganalysis	3
4.1	Block diagram of the proposed system	20
4.2	The flow chart of the PSO algorithm is shown in figure	24
4.3	Flowchart of the proposed steganalysis classification model	32
5.1	Cover image taken from *BOSS* database	34
5.2	Stego image produced after data hiding using *HUGO* algorithm based steganography	34
5.3	Difference image showing change in pixel values after stegnography where white pixels showing changing locations	35
5.4	Steganalysis accuracy for *CC-PEV* features (Features selected=80)	37
5.5	Steganalysis sensitivity for *CC-PEV* features using *APSO* (Features selected=80)	38
5.6	Steganalysis specificity for *CC-PEV* features using *APSO* (Features selected=80)	38
5.7	Steganalysis sensitivity for *CC-PEV* features using *MI* and *APSO* (Features selected=80)	39
5.8	Steganalysis specificity for *CC-PEV* features using *MI* and *APSO* (Features selected=80)	39
5.9	Steganalysis accuracy for *CC-PEV* features (Features selected=100)	41
5.10	Steganalysis sensitivity for *CC-PEV* features using *APSO* (Features selected=100)	42
5.11	Steganalysis specificity for *CC-PEV* features using *APSO* (Features selected=100)	42

5.12 Steganalysis sensitivity for *CC-PEV* features using *MI* and *APSO* (Features selected=100) 43
5.13 Steganalysis specificity for *CC-PEV* features using *MI* and *APSO* (Features selected=100) 43
5.14 Steganalysis accuracy for *SPAM* features (Features selected=80) . 46
5.15 Steganalysis sensitivity for *SPAM* features using *APSO* (Features selected=80) 47
5.16 Steganalysis specificity for *SPAM* features using *APSO* (Features selected=80) 47
5.17 Steganalysis sensitivity for *SPAM* features using *MI* and *APSO* (Features selected=80) 48
5.18 Steganalysis specificity for *SPAM* features using *MI* and *APSO* (Features selected=80) 48
5.19 Steganalysis accuracy for *SPAM* features (Features selected=100) . 49
5.20 Steganalysis sensitivity for *SPAM* features using *APSO* (Features selected=100) 50
5.21 Steganalysis specificity for *SPAM* features using *APSO* (Features selected=100) 50
5.22 Steganalysis sensitivity for *SPAM* features using *MI* and *APSO* (Features selected=100) 51
5.23 Steganalysis specificity for *SPAM* features using *MI* and *APSO* (Features selected=100) 51

List of Tables

Table No.	Title	Page No.
2.1	Summary of recent techniques	16
4.1	PSO Parameters Selection	25
5.1	Confusion matrix for two class classifier	35
5.2	Performance Evaluation by using *CC-PEV+PSO* and *CC-PEV+MI+PSO* with 80 selected features	36
5.3	Performance Evaluation by using *CC-PEV+APSO* and *CC-PEV+MI+APSO* with 80 selected features	37
5.4	Steganalysis Accuracy for *CC-PEV* with 80 selected feature	40
5.5	Performance Evaluation by using *CC-PEV+PSO* and *CC-PEV+MI+PSO* with 100 selected feature	40
5.6	Performance Evaluation by using *CC-PEV+APSO* and *CC-PEV+MI+APSO* with 100 selected feature	41
5.7	Steganalysis Accuracy for *CC-PEV* with 100 selected feature	41
5.8	Performance Evaluation by using *SPAM+PSO* and *SPAM+MI+PSO* with 80 selected features	44
5.9	Performance Evaluation by using *SPAM+APSO* and *SPAM+MI+APSO* with 80 selected features	44
5.10	Steganalysis Accuracy for *SPAM* with 80 selected feature	45
5.11	Performance Evaluation by using *SPAM+PSO* and *SPAM+MI+PSO* with 100 selected features	46
5.12	Performance Evaluation by using *SPAM+APSO* and *SPAM+MI+APSO* with 100 selected features	49
5.13	Steganalysis Accuracy for *SPAM* with 100 selected features	49

List of Abbreviations

ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
APSO	Adaptive Inertia-weight Particle Swarm Optimization
AUC	Area Under Curve
BBC	British Broadcasting Corporation
D-AllCNN	Discretized-All Condensed Nearest Neighbour
DCT	Discrete Cosine Transform
DT	Decision Trees
DWT	Discrete Wavelet Transform
FLD	Fisher Linear Discriminant
GA	Genetic Algorithm
GARCH	Generalized Autoregressive Conditional Heteroskedasticity
GMM	Gaussian Mixture Model
HUGO	Highly Undetectable Steganography
IQM	Image Quality Metrices
k-NN	k-Nearest Neighbor
LRC	Logistic Regression Classifier
LSB	Least Significant Bit
MI	Mutual Information
NB	Naives Bayes
PSO	Particle Swarm Optimization
RFE	Recursive Feature Elimination
RGB	Red, Blue, Green
RISAB	Steganalysis of Image based on Region using Artificial Bee Colony
ROC	Receiver Operating Characteristics
PCA	Principal Component Analysis
PSO	Particle Swarm Optimization
RBF	Radial Basis Function
REP	Reduced Error Pruning

ROC	Receiver Operating Characteristics
SD	Statistical Difference
SPAM	Subtractive Pixel Adjacency Matrix
SVM	Support Vector Machine
WFLD	Weighted Fisher Linear Discriminant
YASS	Yet Another Steganography Scheme

Chapter 1

Introduction

1.1 Steganography

In recent years, data hiding has become popular area of research. Cryptography was mainly used in the early days for secure communication. However, only encrypted information is not sufficiently secure, and this is the reason why the hidden information has been arisen. Steganography is one branch of data hiding. It is the art to hide and transmit the data through a carrier. In addition to the sender of the message and the recipient, no one awares about the existence of the message, so it protects data from unauthorized or unwanted visions. Steganography has become a digital strategy to hide the data in some types of multimedia, like pictures, audio files (for example, in .wav or mp3 format) and in a video file. The general framework of steganography is shown in figure 1.1. It shows that the message is embedded in the cover image by using stego key and embedding process to form stego image.

There should be no obvious differences between the two given images, *i.e.* stego image and the original image while embedding the hidden message. The Steganographic process is considered to be safe if there are no detectable artifacts in the stego due to the incorporation of the message. The statistical properties of the cover images set and the stego images set should be similar. The steganography system is considered damaged if any of the existing algorithm can predict if a particular image contains an embedded data with a accuracy better than the random guessing. For a more precise treatment of the steganographic security concept, refer to [1] [2]. Steganography aims to allow hidden communications by incorporating hidden information into digital medium and the invisibility of the hidden message. There is a high potential to exploit steganography for hidden transmission of data: for example, it has been discovered by a recent spy issue that smart government agency widely use steganography. Effective counter-measures for steganography are greatly needs to take place for different purposes.

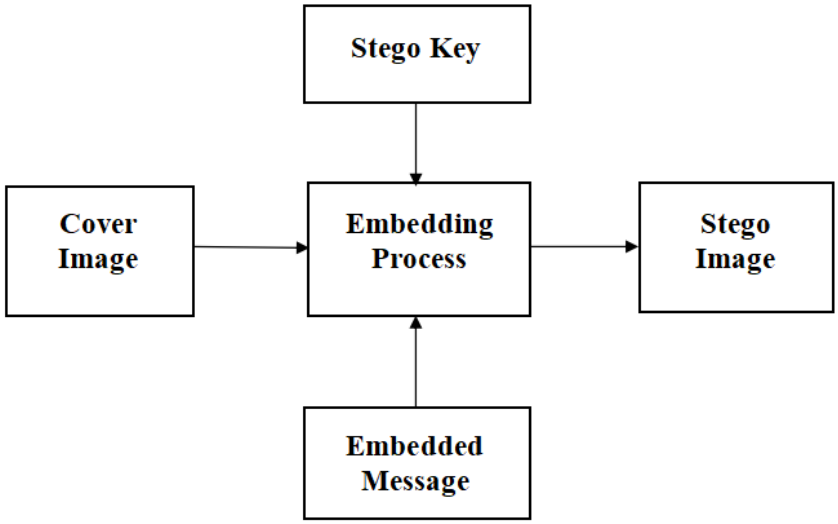

Figure 1.1: Block diagram of Steganography

1.2 Steganalysis

Although steganography can provide a safe means of communicating with government and businesses, but if used by terrorists or criminals, it could from suffer serious consequences. On the contrary, steganalysis was proposed to determine if there are some secret messages inserted in the image or video. These days, steganalysis is a field of active research because of the high number of digital media acting as coverage signals and the accessibility of the social connection network. By incorporating information into an innocuous cover medium secretly, the transmitter waits for the message to reach to the receiver without suspect.

Steganalysis is the skill to discover the un-revealed information from multimedia file being unfamiliar of the details of the existing steganographic information. It aims to discriminate between cover objects and stego objects. The art of steganalysis is always more important in computer forensics, to detect and track suspected criminal activities and security information to avoid the escape of unauthorized data. Steganalysis can also be used to evaluate the weaknesses of the steganographic algorithms. In addition, many open-source applications are easily accessible. [3] [4].

The general framework of steganalysis is shown in figure 1.2. It shows that when we input stego image and cover image into an extracting process, it provides output of whether an image is stego or cover.

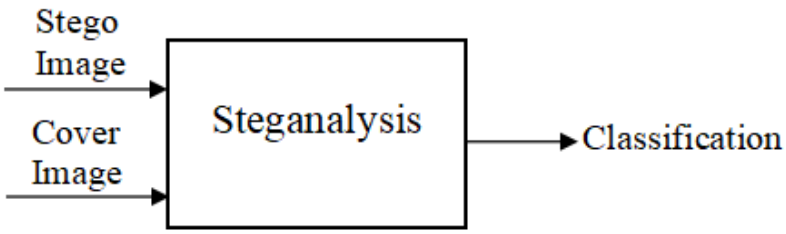

Figure 1.2: Block diagram of Steganalysis

1.3 Approaches of Steganalysis

Basically, Steganalysis approach can be divided into two categories: i. Static ii. Dynamic.

1.3.1 Static Steganalysis

The objectives of the static steganalysis are to discover the existence or inexistence of a hidden message and to identify the algorithm utilized to embed the secret data into the cover medium.

1.3.2 Dynamic Steganalysis

In Dynamic steganalysis, the purpose is to know the length and position (s) of the embedded message, the secret key used in the process of inputting the message, some parameters of the stereo-implanting algorithm and the discovery of the hidden message.

1.4 Classification of Steganalysis

Steganalysis can be broadly divided into two groups, named as, specific and universal approach.

i. Specific Steganalysis The objective of specific steganalysis is to first focus on the analysis of the steganographic algorithm used for the embedding process, so the approach is considered in relation to the steganographic algorithm. The next step includes finding such characteristics of the original image that have changed due to data embedding procedure. Thorough information about the steganographic algorithm is required to design the specific steganalysis algorithms. In other words, the process mainly focuses on a certain type of steganographic algorithm that is used to embed messages.

ii. Universal Steganalysis Universal steganalysis do not detect the certain steganographic method, but it causes the precise identification of the detection of the medium that may contain secret message. Universal steganalysis is more practical as compared to specific steganalysis, because it is independent on the steganographic algorithm.

1.5 Universal Steganalysis

Universal detection aims to classify images provided in two classes: cover and stego images. Some of the existing universal steganalysis techniques extracts some image features, followed by selection and design of a classifier. After designing the classifier,it is trained by using the features extracted from the training images and finally classify the features.

Universal image steganalysis consists of the following main steps:

i. Image pretreatment: It is the process of Performing operations for the considered images before extracting features, like transforming Red, Green, Blue *(RGB)* images into grayscale, cropping, JPEG compression, Discrete Cosine Transform *(DCT)* or Discrete Wavelet Transform *(DWT)* conversion and so on, in order to enhance the classification capability.

ii. Feature extraction: In this phase, informative features are extracted. The

features extracted from the images must be sensitive to insertion or modification. The selected features feature vectors should be of low dimensions, so that the computation complexity of the training and classification can be decreased.

iii. Feature Selection: It is the process of selection of a subset of relevant or important features. It is engaged to obtain the highest accuracy of classification.

iv. Classifier selection and design: In this phase, suitable classifiers are designed on the basis of the selected attributes. Series of images are taken for training of classifiers and key parameters of classifiers are obtained that are to be used in the subsequent classification process.

v. Classification: In the end, the deduced classifier is exploited to distinguish the given images into their classes.

1.6 Features Extraction

In feature extraction technique, relevant features which are sensitive to data insertion are extracted. Features must be of low dimension, which reduce the complexity of training process. They must be sensitive to the data insertion process. In other words, the cover image features and the stego-image features must be completely different. The superior difference among the featurs reveals that the features are informative. The features should be as common as possible, i.e. they are effective on different kinds of images and different data-hiding systems. To map a high-dimensional input image to a low-dimensional feature space, feature extraction phase is needed. A well-trained classification is obtained after the training phase. The classifier constructs the decision boudaries to allocate the feature space to positive sections (the stego image) and negative sections (the cover image) using the created feature vectors, that are extracted from the images of training. The number of common features of universal steganalysis are explained as follows.

i. Image Quality Metrices *(IQM)*: Most steganographic approaches can distort the stego image. The objective of the *(IQM)* is to provide a quantitative metric on the basis of the characteristics of the image, to investigate the distortion of the image features. The statistical evidence obtained through steganographic approaches can be identified with the *IQM* set and can also be used for detection.

Choosing the IQM determines the accuracy of the detection.

ii. Moment Based Features: It can be considered that the impact of the approach to the steganographic image presents some disturbances in the cover image. Due to the introduction of noise, some statistical properties of the cover images can be changed. These changes strongly revealed in the wavelet domain.

iii. Correlation Based Features: The local correlation model image can be changed after the data input process. Correlation can be defined as the inter-pixel dependency of the spatial image and the coefficient dependency of intra-block or inter-block DCT of *JPEG* image correlation.

1.7 Feature Selection

Feature selection techniques attempt to identify important features and eliminate the entire set of as insignificant or redundant feature. The process of feature Selection searches for the best subset of features that may affect the entire dataset by minimizing the loss of information.

Feature selection methods can be categorized into following categories:

i. Filter methods: The filter process is used before the classification process; therefore, they do not depend on the classification algorithm used. For each feature, the weight value is calculated so that the parameters having better weight values are selected for representation of the original data set.

ii. Wrapper methods produce a set of candidate properties by addition and elimination of features to create a feature subset. Therefore, they use accuracy to estimate the resulting feature set. The performance of Wrapper methods are generally better than the filtering methods.

Merits to implement selection process of features in steganalysis are:

i. Inefficient features are eliminated and the informative features are selected to further use it in classification process.

ii. Accuracy of classification of the classifier can be improved by implementing this phase.

iii. By adopting this technique, selected features can assist to notice the sensitive features to an arranged pattern of steganography in the training phase of the clas-

sifier.

iv. Another advantage of feature selection is the reduction of complexity of computation for both to train the classifier and to extract the features. Hence, it is essential to reduce the dimension of features by eliminating redundant or irrelevant features and selecting the relevant ones.

Chapter 2

Literature survey

In this chapter, feature extraction and feature selection techniques used in steganalysis has been reviwed in details.

2.1 Feature Extraction Techniques:

There are various techniques proposed in steganalysis problem to extract features like *CHEN* having 486 feature components [5], *CC-CHEN* having 972 components [6], *SPAM* having 686 components [7], *LIU* having 216 components [8], *CC-C300* having 48,600 components and *CFn* having 7850 components [9].

In [7] Bas displayed a novel way to deal with steganalysis of embedding techniques that inserts data in spatial domain by using the way that the clamor segment of commonplace advanced media shows short-extend environments while the stego commotion is a free irregular part ordinarily not found in computerized media. The differences between adjacent pixels were demonstrated as a first order and second order Markov chain. To obtain the element vector for the process of steganalysis, empirical probability transition matrix is calculated. On the basis of detection of Least Significant Bit *(LSB)* Matching that was performed on four different datasets of images, the evaluation was made. For the purpose of classification, the Support Vector Machine *(SVM)* was used.

In [10] an enhanced approach was proposed by Liu in view of neighboring joint thickness to distinguish an all-around composed versatile steganography in DCT area, having extraordinarily enhanced before DCT-inserting expressions. They additionally proposed another way to deal with steganalysis of Yet Another Steganographic Scheme *(YASS)*, by contrasting the neighboring joint thickness of all hosts blocks that were utilized for information implanting and the non-applicant neighboring squares. For classification, *(SVM)* and Logistic Regression Classifier *(LRC)* were utilized. It is demonstrated that, in steganalysis of DCT-implanting based versatile steganography, their approach has increased impressive great recognition

execution contrasted with the past *JPEG* steganalysis techniques.

In [8], Fridrich proposed an ensemble classifier as an alternative to the complex classifier, *SVM*. To detect the dependencies among the cover features, firstly, high-dimension selected pre-features are put together. Afterwards, a group of weak classifiers is constructed on arbitrary subspaces of the prefeature space. Then after combining the decisions of the individual classifiers, a final classifier was built This technique worked exceptionally well for the Highly Undetectable Steganography *(HUGO)* and nsF5 calculation.

Holub *et al.*[9] proposed an ensemble classifiers worked with the combination decisions of powerless and unsteady base learners executed as the Random Forests. The training complexity of the ensemble scales substantially more positively permitting to deal with high-dimensional component spaces and vast preparing data, evacuating accordingly the restrictions forced by the accessible computing assets that have frequently controlled the detector design before. Execution wise, ensemble classifiers offer accuracy equivalent to, and regularly far and away superior to, the substantially more perplexing *SVMs* at a small amount of the computational cost.

Cho *et al.* [11] presented an image steganalysis framework based on blocks, and broad execution assessment of square based image steganalysis was led. The proposed strategy. Firstly images are divided into blocks, then multiple classes are assigned to image blocks. Results of individual blocks are combined via decision fusion to perform steganalysis. The execution of the proposed technique is analyzed to be less sensitive to the decision combination techniques yet highly sensible to the decision of classifier. In particular, the Fisher Linear Discriminant *(FLD)* classifier outperform the linear Bayes by a substantial margin.

Hou *et al.* [12] proposed a steganalysis system in view of Gaussian Mixture Model *(GMM)* grouping. First of all, in the training phase, training samples are classified into the limited groups independently by using *GMM* clustering and after that steganalizers are built for all the categories. While in the testing phase, a measurement called posterior probability is calculated for all the testing samples and samples having maximum posterior probability is fed to steganalyzer for testing. They demonstrate that the discovery execution of the structure proposed in

their investigation is more than other steganalysis system specifically prepared on a varied image dataset. Also, the proposed system takes care of the steganalysis issue of countless sets, which mitigates the weight of classifier.

Pathak *et al.*[13] displayed a strategy in which the dataset from Berkley's image dataset BSD300 was utilized in its unique arrangement with no alterations and from the three spaces, features were extricated and classified by using *SVM*. The insights in three domain areas, *i.e.* frequency, wavelet and spatial areas are gotten and values of factual component were ascertained. At that point every one of the features is assembled for preparing in *SVM* to obtain a prepared model. Test images after testing with the prepared *SVM* were classified into given categories. Another steganalysis technique was proposed in [14] utilizing a direct number of highlights. The feature set is the mix of both generalized autoregressive conditional heteroskedasticity *(GARCH)* display characteristics and statistics of higher order. It has been utilized to appropriately show the overwhelming followed circulation of non-inexact wavelet. These features are removed from change space to follow the prerequisites of a blind steganalyis technique. The proposed highlight based steganalysis technique outflanks best in other plans while utilizing features of similar order.

Kong *et al.* [15] proposed a multi-order feature arrangement method for jumbled steganalysis by adapting new component portrayals, which can adjust both low and higher order features amongst preparing and testing sets iteratively. They conduct a few crisscrossed investigations on both the open image database and research facility picture database and contrast their approach with past expressions. They delineate that their technique can enhance the execution for mismatched steganalysis.

In [16], Sajedi proposed a novel steganalysis procedure to expand the precision of steganalysis images. In the wake of finding the utilized steganography strategy, a reasonable steganalysis model was utilized. In such manner, to create fuzzy guidelines from elements of the stego images, an evolutionary fuzzy calculation was proposed. As indicated by the results, their approach increase the discovery rate of steganalyzers contrasted with the traditional utilization of steganalysis techniques. The benefit of their given technique is that in the occurence of new

strategy of steganography, the fuzzy rule base can be overhauled and the given pattern can be utilized for the steganalysis.

In [17], Li proposed to utilize the measurements of some features having new shapes as contributions for $3D$ steganalyzers. They investigate neighborhood features by applying different combinations utilized for $3D$ steganalysis by assessing their importance to the class name and by testing their execution in the trials. They demonstrate that the presented $3D$ element vector gives the improved outcomes to the steganalysis of six $3D$ data concealing calculations.

A steganalytic scheme was proposed in [18] by Feng to distinguish content-versatile information hiding in binary image and they investigated the insertion effect presented by inserting methodology on the basis of l-shape design. A 2-dimensional list of features was planned. Further, 28 classes of patterns that contain one or part of an l-shape design are considered to demonstrate the implanting impact related with l-shape designs. In light of the execution evaluation on these pattern classes, a 32-dimensional elements vector set that comprises of the appropriation of 4 classes of 43 estimated designs is proposed.

In [19], Christaline exhibited a method in which featuress are extricated domain to create a rich dataset from the noise elements specifically in spatial. Six classifications of residuals are separated including the quick neighbors, second request direct quadratic model and third request straight quadratic model. As the process of image steganalysis is a two class advanced characterization issue, this exploration has actualized individual and combination classifiers to accomplish most extreme order precision.

In [20], Karampidis gave a definite report of different techniques proposed for steganalysis applicable to digital images. The proposed techniques center to the inserting method and endeavor to discover picture features or measurements changed by the installing calculation. So, this steganalysis approach has phenomenal accuracy just when performed on the particular steganographic calculation, however even a little change in the insertion process for the most part results to low steganalysis precision. Consequently, universal or blind steganalysis is utilized. Such strategies can recognize secret information in any case the steganographic procedure that were inserted to the computerized image.

2.2 Feature Selection Techniques:

Feature selection techniques attempt to identify important features and eliminate the entire set of an insignificant or redundant feature.

In [21] Geetha exhibited the utilization of Markov Blanket-Embedded Genetic Algorithm *(MBEGA)* for dimensionality lessening by selecting features in the system of steganalysis. The quality of the framework has been shown on up to 1750 images. To improve the solution of feature selection and search space, addition and deletion of features selected from Genetic Algorithm *(GA)* are done by memetic operators based on embedded markov blanket. The evaluation results demonstrated that the given technique is efficient in selection of features, cost of computation and accuracy of classification.

Adeli *et al.* [22] present the utilization of *GA* to select significant and informative features by using a novel fitness function called The Area Under the receiver operating characteristics Curve *(AUC)*. The execution of *k-NN* calculation is enhanced by the point of this selection of features. The characterization execution of K-Nearest Neighbours *k-NN* calculation in examination with alternate classifiers (for example, C4.5, *SVM*, and Relief) is considerably enhanced by the proposed technique.

Zhang *et al.* [23] displayed an ideal feature selection utilizing Particle Swarm Optimization *(PSO)* calculation for image steganalysis. The classification activity incorporates a *SVM* classifier, which works with *PSO* and serves to diminish the quantity of features and to build the discovery execution. The execution of the proposed method is promising and show that the chosen features can dependably recognize. They show that their proposed cross hybrid calculation diminishes classifiers training complexity as well as increment the right classification rate.

Another strategy for feature selection was introduced in [24]. The proposed thought utilizes *PSO* with fitness function keeping in mind the end goal to allocate low weights to insignificant features while instructive features given higher weights. The fitness function of the particles was considered as *AUC*. They assert

that the proposed technique can enhance the performance of classification of the *k-NN* calculation in examination with the other essential strategy in domain of highlight weighting, for example, Tabu Search, *GA*, Mutual Information*(MI)*, and chi-squared.

In [25] a novel optimization algorithm in view of *PSO* approach in unique situations was proposed, in which a few mechanisms were utilized to vanquish difficulties and necessities of dynamic conditions. To expand diversity of the particles in the proposed calculation, a novel technique in view of change in speed vector and molecule positions was introduced and so as to enhance the effectiveness of the calculation, a local search in view of versatile exploiter molecule around the best discovered position was recommended. At last, a mechanism has been exhibited to focus on the global optimum peak.

In [26], a feature selection technique is proposed for steganalysis called *IFAB* was proposed by Mohammadi to identify the cover images and the stego images using Artificial Bee Colony *(ABC)* based feature selection with *SVM* classifier. The obtained results demonstrate that their technique is easy and efficient for performing the task of steganography.

A steganalytic technique of feature selection in view of the Fisher criterion utilized as a part of pattern recognition was proposed in [27] by Lu, in which the detachability of single-measurement and numerous measurement features, joined with estimation of the Euclidean distance, is investigated. They demonstrate that the proposed strategy can viably diminish the features while protecting steganalytic precision, and can likewise significantly enhance the steganalytic productivity.

Wu *et al.* [28] described that a semi-managed learning calculation which coordinates weighted Fisher linear discriminant *(WFLD)* and K-implies grouping into a join structure is contrived to solve unbalanced image steganalysis. The K-means clustering is utilized to create class names and *WFLD* for selection of subspace. The unequal preparing set is adjusted utilizing the multiview coordinate resampling strategy to pick certain stego pictures from unlabeled illustrations. The proposed technique can viably recognize *MBs* and *nsF5* steganographic strategies and beat existing steganalysis approaches.

In [29], Megas introduced a novel unsupervised steganalysis strategy. Utilizing the

proposed approach shows superior performance than the state-of-the-art strategies. Evacuating the need of a training data set is the significant commitment of their work. The proposed approach has been tried utilizing three stegano-realistic techniques: *HUGO*, *LSB* coordinating and *WOW*. It is demonstrated that better grouping precision can be accomplished than that got utilizing traditional supervised steganalysis (Rich Models, Ensemble Classifiers and SVM). The proposed strategy gives striking execution regardless of whether the pictures are chosen unevenly from various databases or if the implanting bit rate is obscure and variable for various stego tests.

Rostami *et al.*[30] proposed a technique to select features based on *PSO* in which AUC was used as a fitness function. The evaluation results proved the efficiency of the proposed method in selecting features.

In [31], Hou examined the highly imbalanced steganalysis with small training samples issue primarily from the point of view of feature selection. They assessed eight diverse component selection measurements with three distinctive grouping calculations on four delegate steganlytic features and found that feature selection with the classifier *FLD* alone can beat the *HISST* issue even for high-dimensional steganalytic feature sets and displays better execution over sampling and new learning technique by and large, and the groupings of different methodologies don't create more enhanced outcomes.

In [32], Mohammadi proposed a Steganalysis of image based on region by using Artificial Bee Colony *(RISAB)* was proposed in which best sub-image is selected using ABC, having the highest density with respect to the changed inserting pixels. Then selected features by *IFAB* are extracted, Both features selected using *IFAB* and features extracted by *RISAB* are combined to train the classifier.

Veena *et al.* [33] proposed a widespread blind quantitative steganalyser with diminished features. The steganalyser chips works on both local and global features. The features are joined utilizing Greedy Randomised Adaptive Search Procedure *(GRASP)* calculation and the lessened training occurrences of the chose connected model are achieved by the proposed Discretized-All Condensed Nearest Neighbour *(D-AllCNN)* strategy. Further, the dimensionality of the element is decreased by Recursive Feature Elimination *(RFE)*. The proposed streamlining is powerful for

both conventional non-versatile and content versatile spatial *LSB* calculations. Adeli *et al.*[34] proposed another strategy for include choice inlight of the Adaptive inertia weight-based *PSO (APSO)* technique to chose the components of feature vectors from the image that are better than others for steganalysis. In this technique, *APSO* is prepared to versatile modifying plan of latency weight through the advancement procedure.

Table 2.1 shows the summary of recent techniques used in the process of steganalysis. In [21] Geetha used *WAM*, *IQM*, Fridric's and higher order statistics features and *MBEGA* algorithm was used for feature selection. Images were classified using *SVM* Classifier and higher detection rate was acheived. In [23], Statistical moments were used as features which were further selected using *PSO*. Accuracy of 88.34% was achieved using *SVM* classifier to classify the images into given classes. In [27], Lu proposed fisher criterion for feature selection process. Mohammadi *et al.* [26] used *SPAM* and *CC-PEV* features for classification. The features were further selected by using *IFAB* technique and classifier *SVM* was utilized to identify the class of given images. Detection rates were 60.98% and 64.6% for *SPAM* and *CC-PEV* features. In [35] accuracy for *SPAM* features was improved using *ABC* for feature selection technique and *kNN* classifier. In [32] detection rates for *SPAM* and *CC-PEV* features were further improved to 68% and 70% respectively, by using *IFAB* and *RISAB* techniques. Sajedi *et al.* [36] used ABC for selecting features of *SPAM* and *CC-PEV*. Accuracy of 66.08% and 69.06% was achieved using *k-NN* as a classifier. In [34], Adeli proposed *APSO* to select the features and the technique shows higher detection rate than the state-of-the-art techniques.

Table 2.1: Summary of recent techniques

Paper	**Feature Extraction**	**Feature selection**	**Data**	**Outcomes**
Geetha. (2010) [21]	*WAM*, *IQM*, Fridrich's and higher order statistics	*MBEGA-SVM*	Natural Images	Accuracy *WAM*(92%) *IQM*(88%) Fridrich's(80%) Higher order statistics(84.4%)
Chen. (2012) [23]	Statistical moments	*PSO-SVM*	CorelDRAW Version 10.0	Accuracy (88.34%)
Mohammadi. (2013) [26]	*SPAM* and *CC-PEV*	*IFAB-SVM*	BOSSbase (version 1.01) Images	Accuracy *SPAM*(60.98%) *CC-PEV*(64.6%)
Mohammadi. (2014) [35]	*SPAM*	*ABC-kNN*	BOSSbase (version 1.01) Images	Accuracy *SPAM*(66.08%)
Mohammadi. (2016) [32]	*SPAM* and *CC-PEV*	*IFAB-RISAB-SVM*	BOSSbase (version 1.01) Images	Accuracy *SPAM*(68%) *CC-PEV*(70%)
Sajedi et al .2017 [36]	*SPAM* and *CC-PEV*	*ABC-kNN*	BOSSbase (version 1.01) and greyscale Images	Accuracy *SPAM*(66.08%) *CC-PEV*(69.06%)
Adeli. (2017) [34]	*SPAM* and *CC-PEV*	*APSO-SVM*	BOSSbase (version 1.01) Images	Accuracy *SPAM*(82.62%) *CC-PEV*(87.72)
Lu. (2013) [27]	*SPAM* and *CC-PEV*	Fisher Criterion	BOSSbase (version 1.01) Images	Accuracy *SPAM*(72.33%) *CC-PEV*(64.25%)

Chapter 3

Problem Statement

3.1 Research Gap

Steganalysis algorithms attempt to recognize stego images from clean images. In well-known steganalysis techniques, the critical issue is to distinguish the presence of the hidden data. To apply this method, the steganalyst needs to separate an arrangement of features from a training data set and prepare a classifier. It is a pivotal pre-processing method for powerful information analysis, where just a subset from the original features is selected to disposed of insignificant or excess features. Utilizing an expansive number of features is unpleasant in terms of computational cost, classification precision, and training time in steganalysis because of the scourge of measurement. So there is a need to diminish computational cost and enhance accuracy of the information analysis process. In order to reduce feature space, meta-heuristic optimization algorithms are much preferred in current research. One such system which has picked up prevalence in the ongoing years is the *PSO*. Howevere, there are some downsides involved with the *PSO*. Due to the stochastic idea of the calculation, *PSO* does not generally ensure the finding of the ideal arrangement each and every time. In certain circumstances, the calculation loses its diversity which effects the execution of the algorithm negatively due to the the quick convergence rate of the algorithm. So, an APSO which involves change in weight parameter during new iteration based on diversity measures can be used for feature selection and pre-selection phase can be introduced to decrease the computation time of *APSO* and to increase the classification accuracy.

3.2 Statement

To manage the continuing growth of number of features utilized as a part of current steganalysis, an adaptive particle swarm optimization algorithm is used, which involves change in weight parameter during new iteration based on diversity measures. The calculation likewise exploits data from the past generations to

adjust to the variations in the surroundings. The various swarms adjust to the new areas and help in following the ideal solution. The calculation is intended to have higher reusability, improved accuracy, quicker convergence, less demanding usage and capacity to work under serious and high recurrence changes. With a specific end goal to additionally enhance the execution of *APSO*, pre-determination stage has been presented in which features sorting is carried out on the basis of Mutual Information*(MI)* and just initial few from entire element vector are fed into the *APSO* calculation. This diminishes the calculation time of the *APSO*.

3.3 Contribution

1. The complexity for both classifier training and feature selection has been reduced by introducing pre-feature selection phase using mutual information method.
2. The meaningless features has been pruned using further *APSO*.
3. The classification of images into two distinct classes- stego and cover, can be done with improved detection accuracy.

Chapter 4

Proposed Technique

4.1 System Module

Present work involves implementation of an image steganlaysis system which can effectively classify a set of images into cover and stego images. *BOSS* [37] dataset has been selected for this research work which contains 10,000 cover images in it. Stego images has been produced from cover images using data embedding by *HUGO* steganography method. Then Subtractive Pixel Adjacency Matrix *(SPAM)* and *CC-PEV* feature set is evaluated from both cover and stego dataset. Pre-feature selection phase has been implemented using mutual information based feature selection method. The features selected above are further reduced based on AUC by a classifier through adaptive *PSO* in which objective is to maximize the AUC parameter based on selected feature columns in each iteration. Finally, selected features using *APSO* are fed into classification system in which *kNN*, Decision Tree*(DT)* and *SVM* are used in order to classify the whole image dataset into cover and stego images. Then performance evaluation has been measured and compared with one another. The basic steps in the proposed system module are given in the form of flowchart in the figure 4.1.

4.2 Feature Extraction

4.2.1 Spatial-domain based feature selection for steganlysis

Pevn et al. [7] pointed out a second order Markov transition probability matrix.The selected threshold value is 3 that can give better results in steganalysis. They concatenated 343-dimensional Markov transition probability matrices by taking the neighboring-pixel-difference matrices in both the horizontal and vertical direction and another 343 neighboring- pixel-difference matrices both major

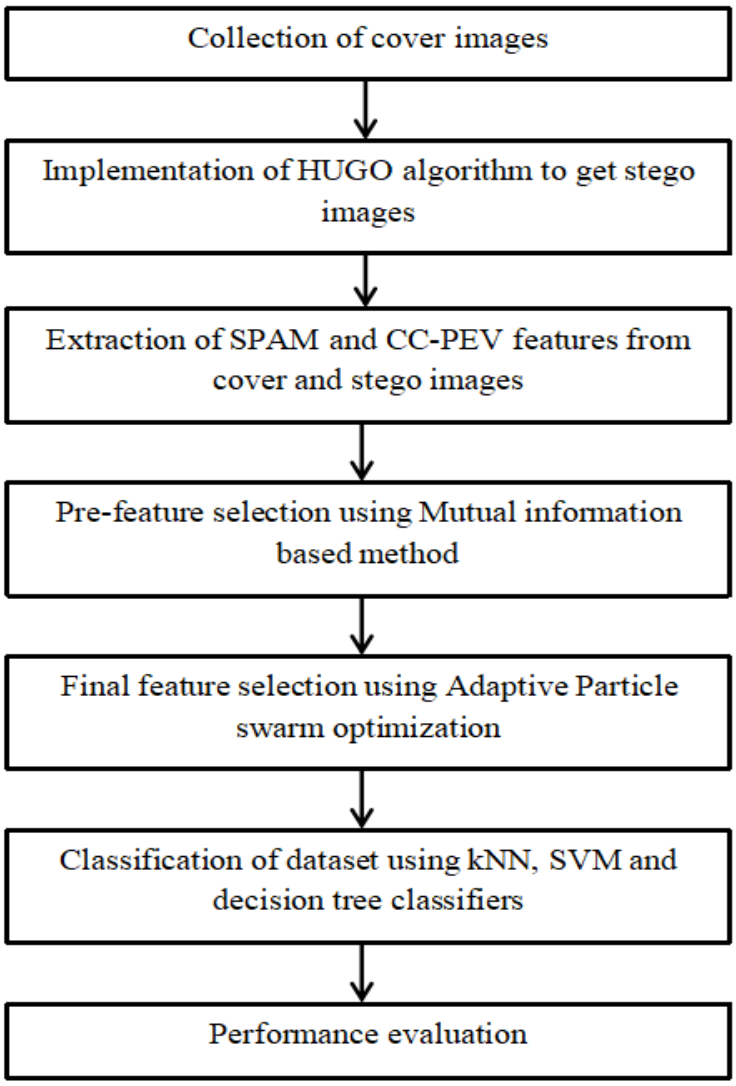

Figure 4.1: Block diagram of the proposed system

diagonally and minor diagonally to get 686 dimensional feature vector named as *SPAM* features. In this work, first *SPAM* features are first extracted from both cover images as well as stego images. Then, *MI* score has been evaluated of every dimension feature vector and sorted the features in descending order. Finally, we used first 120 features for further feature selection with the use of *APSO* when number of selection is set to 80 features. As results are carried out using selection of 100 features as well, in that case 150 features has been selected from the *MI* based feature sorted matrix.

4.2.2 Frequency-domain based feature selection for steganlysis

CC-PEV feature set is a 274 dimensional feature matrix which evolves from the calculation of histogram-coefficients, blocked based artifacts and Markov transition probability matrices that can be evaluated form the detected image data and calibrated image data. When these images are merged it comes out as 548 dimensional feature vector which has been used further for steganalysis. In proposed work, *CC-PEV* features are extracted from both cover as well as stego image dataset. Then, *MI* has been evaluated from the single-dimension features, and re-ordered the features in descending order. Finally, we used first 120 features of the feature components for further feature selection with the use of *APSO*.

4.3 Statistical Dependency and Mutual information between features and labels

The objective of the statistical dependency *(SD)* technique is essentially to quantify whether the estimations of a component are subject to the related class names. Each component is first quantized into one of the QS levels, where the element particular quantization scale is adaptively decided with the end goal that each bin will contain around an equivalent measure of tests over the whole data. The bins are picked along these lines, rather than an ordinary uniform quantization scale, keeping in mind the end goal to loan some factual legitimacy to the event of various quantization levels. The statistical dependence between the discrete features y and the class names z is assessed by the equation.

$$SD = \sum_{y \epsilon Y} \sum_{z \epsilon Z} p(y,z) \frac{p(y,z)}{p(y)p(z)} \tag{4.1}$$

The greater the *SD*, the higher is the reliance between the component features and the class names. For the situation that the element is completely autonomous of the class names, the *SD* will acquire the minimum estimation of 1. In this work, the similarity of this measure is carried out with *(MI)*. *MI* is amount of information that one random variable has about another variable. The MI between the discrete

features y and the class name z is assessed by

$$MI = \sum_{y \epsilon Y} \sum_{z \epsilon Z} p(y,z) \log \frac{p(y,z)}{p(y)p(z)} \tag{4.2}$$

4.4 Particle Swarm Optimization

J. Kennedy and R. Eberhart in 1995 [38] introduced the concept of Particle Swarm Optimization *(PSO)*. At first, continuous nonlinear functions were solved by using *PSO*. The basic idea came from bird-flocking. Consider a group of birds searching the n-dimensional space for its food and no ones knows where it is in the start but they know about the bird position which is close to the food. Hence all the rest followed the best bird nearer to the food. Hence *PSO* considers each bird as a particle in which its position can be given as

$$x_i = (x_{i1}, x_{i2}, x_{i3} \cdots x_{in}) \tag{4.3}$$

Initial solution are selected on random basis and then it tries to converges to the optimum solution after every new iteration. The objective function used to find the best optimum is called as fitness function which varies from one application to another. Also velocity of the particles needs to be evaluated in this which is represented as below

$$v_i = (v_{i1}, v_{i2}, v_{i3} \cdots v_{in}) \tag{4.4}$$

It relates itself to the previous velocity best global and local known positions. It indicated the direction of the particle needed in the next iteration. It can have positive as well negative values. The local best is the best optimum solution in the current iteration whereas global best is the best optimum solution among all iteration till now. The inertia of velocity, best local and global known positions of the corresponding velocity represent the co-operation and competing mechanism in *PSO*. Like in *GA*, it also starts with random solutions which keeps on updating in every solution. The difference between them is that *PSO* use the history data other than crossover and mutation used in genetic algorithm.

The new updated velocity in *PSO* is given as under:

$$v_{d+1} = k^*(w^* v_d + \phi_1 \cdot rand()^*(p_{best} - x_d) + \phi_2 \cdot rand()^*(g_{best} - x_d)) \tag{4.5}$$

$$v_{d+1} = x_d + v_{d+1} \tag{4.6}$$

where w is the inertia weight factor, k is the constriction factor rand () is a random value between 0 and 1 and ϕ_1 and ϕ_2 are acceleration factors.

Acceleration factor decides the step size of the particle in the coming iteration. With too small value of it, the particles do not have enough velocity value so that it cant reach to the target variables. If value is too high, it can over pass the optimum solution. Hence Acceleration factor can be decided by considering the drawbacks of too low and high values. Optimal selection of acceleration factor can reduce the computation time as it does not stick in local minima. Also there is constraint on maximum velocity that can be achieved by the particles which is represented as v_{max}. If a particle has more velocity than v_{max} then v_{max} is assigned to that particle. Value of v_{max} can vary from application to application. The flow chart of the *PSO* algorithm is shown in figure 4.2.

PSO has some advantages over other optimization algorithms like *(GA)* [47]:

i. Cooperation and competing mechanism is used to get the direction of the particle in coming iteration.

ii. Very few parameters are needed in *PSO* as compared to *GA*.

iii. The computing speed of *PSO* does not affected by the complex objective functions.

iv. *PSO* can be used to large number of applications.

4.4.1 Parameters Selection and diversity measure in *PSO*

The PSO parameters Selection is listed in Table 4.1. Programmers may change these parameters according to their problem.

The diversity parameters in *PSO* plays major role in convergence to optimal solu-

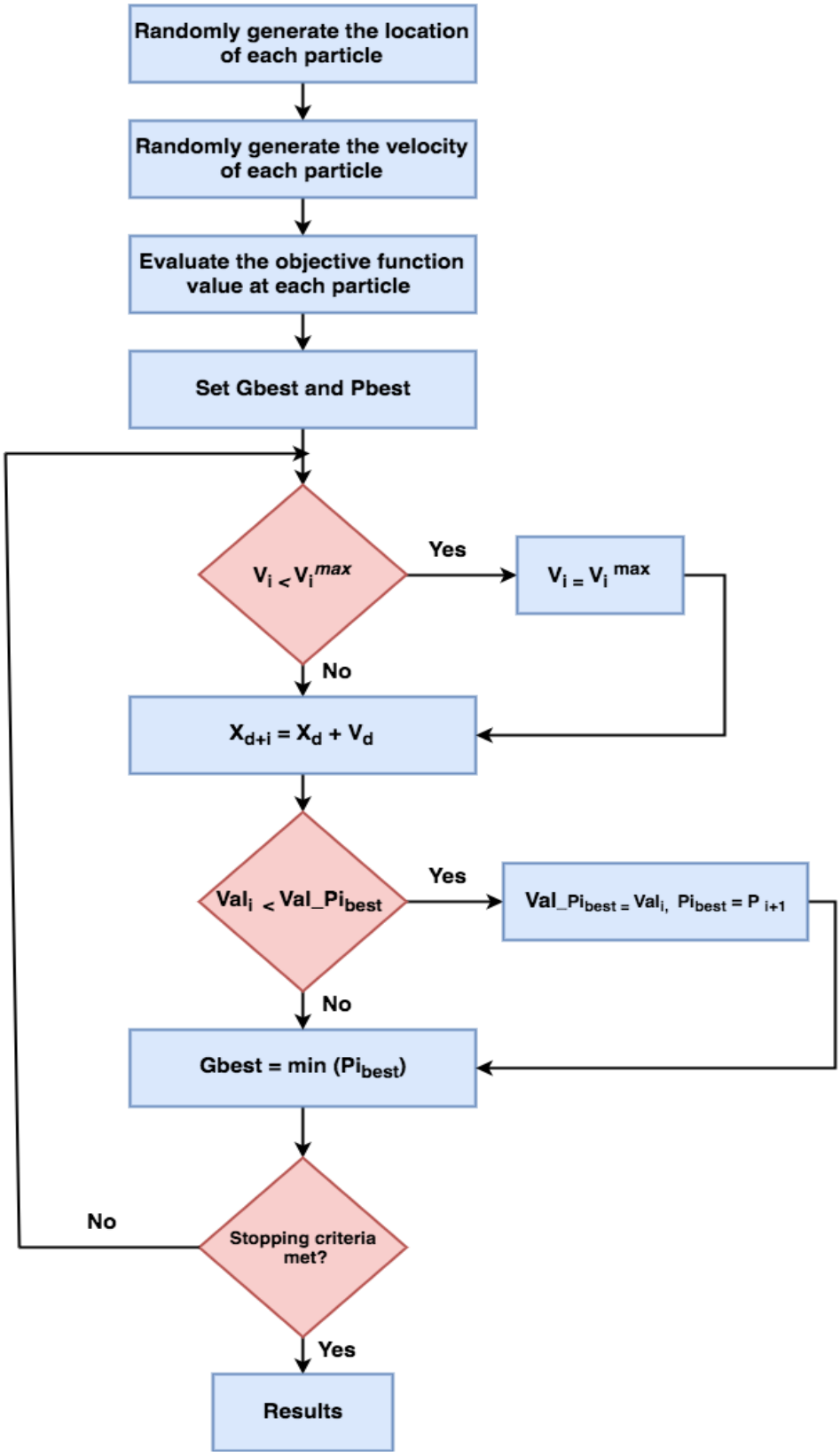

Figure 4.2: The flow chart of the PSO algorithm is shown in figure

tion. Large diversity means that a large searching space is explored by the particles and a smaller population diversity values means the lesser area researched by the

Table 4.1: PSO Parameters Selection

Size of the particles	Depends upon the dimension of the input feature set and increase with the increase in dimension. Twenty to forty were often used in many applications
Dimension	Depends upon the number of control varaibles
Domain	Upper and lower bound constraints decides the domains of the particles
Acceleration-factor	Between two and four
Criteria to stop the optimization process	Maximum iterations selected in the start. Difference of current best and previous best solutions. If there is no improvement in fitness values, than process can be stopped after some pre-set limits.

particles. It is quite significant to compute the search behavior of a *PSO* algorithm when swarm diversity is considered in the search as in the Attractive and Repulsive *PSO* developed in [39]. Therefore, such algorithms needs to accurate measuring of search behavior in swarm with respect of one time to the another time.

MEASURES OF SWARM DIVERSITY

The diversity measures considered in this work are given in this sub-sections.

A. The Swarm Diameter

The maximum distance between any two particles in the swarm is considered as the swarm diameter. The diameter is calculated as follows:

$$|D_{dim}| = \max_{(i \neq j) \epsilon [1, |S|])} (\sqrt{\sum_{K=1}^{I} (x_{ik} - x_{jk})^2}) \tag{4.7}$$

I reveals the dimension of the problem (and its solution), where $|S|$ is the swarm size and x_{ik} is considered as *k-th* dimension of the *i-th* particle position.

B. The Average Distance around the Swarm Center

This measure is calculated as:

$$D_s = \frac{1}{|S|} \sum_{i=1}^{|S|} \sqrt{\sum_{K=1}^{I} (x_{ik} - \bar{x}_k)^2} \tag{4.8}$$

C. Swarm Coherence

Swarm coherence is given by:

$$S_c = \frac{v_s}{\bar{v}} \tag{4.9}$$

Where v_s is considered as the speed of the swarm center given by:

$$v_s = \frac{1}{|S|} \| \sum_{i=1}^{|S|} \bar{v_i} \|_2 \tag{4.10}$$

and $\bar{v}$ represents the average particle speed of the swarm which is formulated as under:

$$\bar{v} = \frac{1}{|S|} \| \sum_{i=1}^{|S|} \bar{v_i} \|_2 \tag{4.11}$$

4.4.2 Adaptive inertia weight-based *PSO*

Inertia acceleration weight needs to be adjusted for every iteration in order to balance exploitation and exploration rate which produced convergence to optimal solution in least number of iterations. High inertia weights enhance the exploration ratios which assist in global search by the particles. Opposite effect happens when inertia weights decrease to small values which reduces the exploration and increase the exploitation and able the *PSO* to stuck in local minima or limits to the local search only. Therefore a balanced value needs to be proposed after every iteration which can be done by introducing diversity measure parameters. In the start, particles do not know much about the space that they explore, therefore they must try to explore the full provided space of the data dimensions and in the global best solution they must addicted to the local search so that effective solution from the

nearby global solution can be achieved. Therefore high inertia rate is needed in the starting iterations and when it tries to converge to the global best the inertia weight need to decrease along with the increase in the iteration of the optimization process. Traditional *PSO* works with the fixed weights which is set in the start, therefore to defend the challenge of varying inertia weights, adaptive *PSO* is used which provides new inertia weight after each iteration based on velocity, position, fitness value of the previous iteration.

To design the adaptive *PSO* system, a random population with number of particles is generated between the range zero an one with D-dimensional space needed in the output which should be equal to total dimension of the input data that needs to explored. To make relation between decisive particles and corresponding feature dimension in order to select or reject it form final solution, a threshold value of the decisive variable need to be considered, if It is high, then corresponding dimension elements are selected and if it less than it is excluded from the selection and selected ones further fed to the fitness evaluation objective function. When initial population is generated , the fitness value is computed for each particles with the help of objective function provided for a particular application. As in this case, fitness function is the AUC value provided by a binary classifier, the higher value of AUC decides the best global solution. Hence this way entire set of particles explore the whole search space to get the best sets of the features in the dataset. After every iteration, all the particles changes their positions and corresponding velocities by experiencing best of each particle and best experiences of the whole swarm. These positions and velocities are updated using equations 4.5-4.11 which provides a new value of inertia weight after every iteration Hence the main difference of adaptive *PSO* from the traditional *PSO* is balancing of inertia weight during coming iterations. The formula adopted by the new inertia weight is given in equation below:

$$w(t) = w(t-1) \times (\frac{1}{D_{dim} + D_s}) + \alpha \times S_v \tag{4.12}$$

where weight(t) is current iteration inertia weight and weight(t-1) is previous iteration inertia weight, alpha is a constant. Remaining parameters are defined

in previous section in diversity measure. Equation 4.12 shows that weight of inertia factor increases when sum of swarm-diameter D_{Dim} and mean difference of particles around the center named as D_s is decreased and average speed of the particles towards the central point is increase which is named as S_V. Variable α have been add to (4.12) in order to balance the effect of S_v on the decrease and increase of the inertia weight. Decrease or increase of mean difference of particles around the center and swarm-diameter D_{Dim} results in opposite effect of increase of decrease of average speed of the particles towards the central point which where is showing the convergence of swarm towards the similar area in the problem space therefore the inertia-weight need to be increased or decreased. Therefore , inertia-weight need to be repetitively adjusted depending upon the variations in diameter of the swarm , the distance of particles around center and the velocity or speed of the particles around the center. Hence these parameters have been considered to adaptively change the inertia weight for the current new iteration.

Hence the motivation behind the usage of adaptive is the concept in the start of the *PSO* algorithm where high exploring rate and low exploitation rate is needed which can be provided by the adaptive inertia weight. Due to high diversity of the particles happens in the initial stage, the low inertia weight is computed in the start, and high value of inertia weight in the end of the optimization which results in adjustment of the exploration and exploitation values, convergence as well as searching steps of *PSO*. By adapting the inertia-weight parameter, the new equation of velocity of the particles is obtained by the following equation:

$$v_{i,j}(t+1) = w(t) \times v_{i,j}(t) + C_1 R^1_{i,j}(Pbest_{i,j}(t) - x_{i,j}(t)) + C_2 R^2_{i,j}(Gbest_j(t) - x_{i,j}(t)) \tag{4.13}$$

After a number of iterations, the best particle is located depending upon the best fitness value provided by AUC of the binary classifier for a two class decisions. As AUC is used as fitness function in presented $APSO$, it is explained as under.

4.4.3 Fitness Function

The selected features are assessed utilizing the AUC. AUC is used as a fitness function of the particles. The Discriminative intensity of binary classifiers are measured by AUC. The selected features are fed into binary classifiers in such a way that targets of class of each row of the feature vector is represented by zero and one for corresponding class *i.e.* in this case stego features are represented by one and cover image features are represented as zero. Then binary classifier trains and validates itself by selecting a ratio from the fed feature matrix. After that it tests the whole dataset which produce a confusion matrix which can be decided from a threshold value as results comes after testing in the range [0 1] and gives True Positive *(TP) i.e.* predicted positive cases that are positive in actual, False Positive *(FP) i.e.* number of predicted positive cases, True Negative *(TN) i.e.* predicted negative cases that are negative in actual and False Negative *(FN) i.e.* number of predicted negative cases that are actual in this way. Particle is the input of this function and scalar value in range of [0 1] is its output. If the value of AUC is high, it means the performance for classfication is remarkable. AUC measure is based on Receiver Operating Characteristic *(ROC)*, a two dimensional curve, to calculate the accuracy of classification of a classifier. Hence ROC curve can be evaluated from above said parameters of the confusion matrix which further depends upon sensitivity and specificity values. The ROC curve is given by sensitivity and 1-specificity values as defined below:

$$Specificity = \frac{TN}{TN + FP} \tag{4.14}$$

By taking integral of the area under ROC curve, we can evaluate AUC parameter. The AUC variable has been depends upon the feature sub-set selected by different particles.

The aim of using AUC measure as a fitness function is to evaluate the performance of the feature vector that has been selected so that it can effectively classify the whole dataset into stego or cover images. In the final step of the iterations, the particle having highest value of AUC value is selected as the final selection of the

features that are provided by *APSO* and can be further fed to the classifiers to categorize the images into given classes.

4.5 Training Step

The last step of the presented technique is to prepare and test a particular machine learning classifier for the undertaking the process of steganalysis. To classify the input features into the given classes is the fundamental objective of the process of steganalysis. So it is necessary to prepare and test a learning approach in light of the removed component vectors of preparing and testing picture sets. In the proposed technique, a few classifiers, for example, *SVM*, Decision Tree(DT), and *k-NN* are utilized in order to train to select feature subset. *SVM* is a managed learning strategy. It plans to build a hyperplane in high-dimensional space with the most extreme edge to arrange input features. When the hyperplane build by *SVM* has the biggest separation to the closest feature focuses from any class, the best consequences of *SVM* are achieved. If in the case, data samples are not directly separable, the samples space are changed high-dimensional space. In this work, *SVM* is utilized with he *rbf* portion to carry out the process of steganalysis. DT standout amongst the other classifiers for classification purpose. The tree produced by training data has internal node which represents decision variable, leaf node which denotes the label of class and the way from the roof to a leaf represents the rules of classification of pattern. *k-NN* is a non-parametric grouping strategy that lone uses the spatial conveyances of observational examples without earlier presumptions about the dispersions of classes. Rather than these three classifiers Naive bayes is utilized as a part of fitness capacity of *APSO*. *NB* is a learning strategy based on the probability that uses the Bayes hypothesis. In simple terms, *NB* classifier assumes that features are conditionally independent of each other given the target class which is used to find the AUC in adaptive particle swarm optimization.

To prepare every machine learning classifier, *APSO* found the best element subset. Calculation is connected all in all dataset to keep up the most noteworthy features

or expel the less effective characteristics. In the determination step, there is a predefined limit, the features having the value higher than this limit will be remained while the features having lower values than the edge will be evacuated. After completing the step of feature selection, A classification process is then applied on the selected subspace of features after completing the process of feature selection. At last,the testing dataset is used to classify the images into given classes in order to assess the proposed model The flowchart of the proposed method is illustrated in Fig. 4.3.

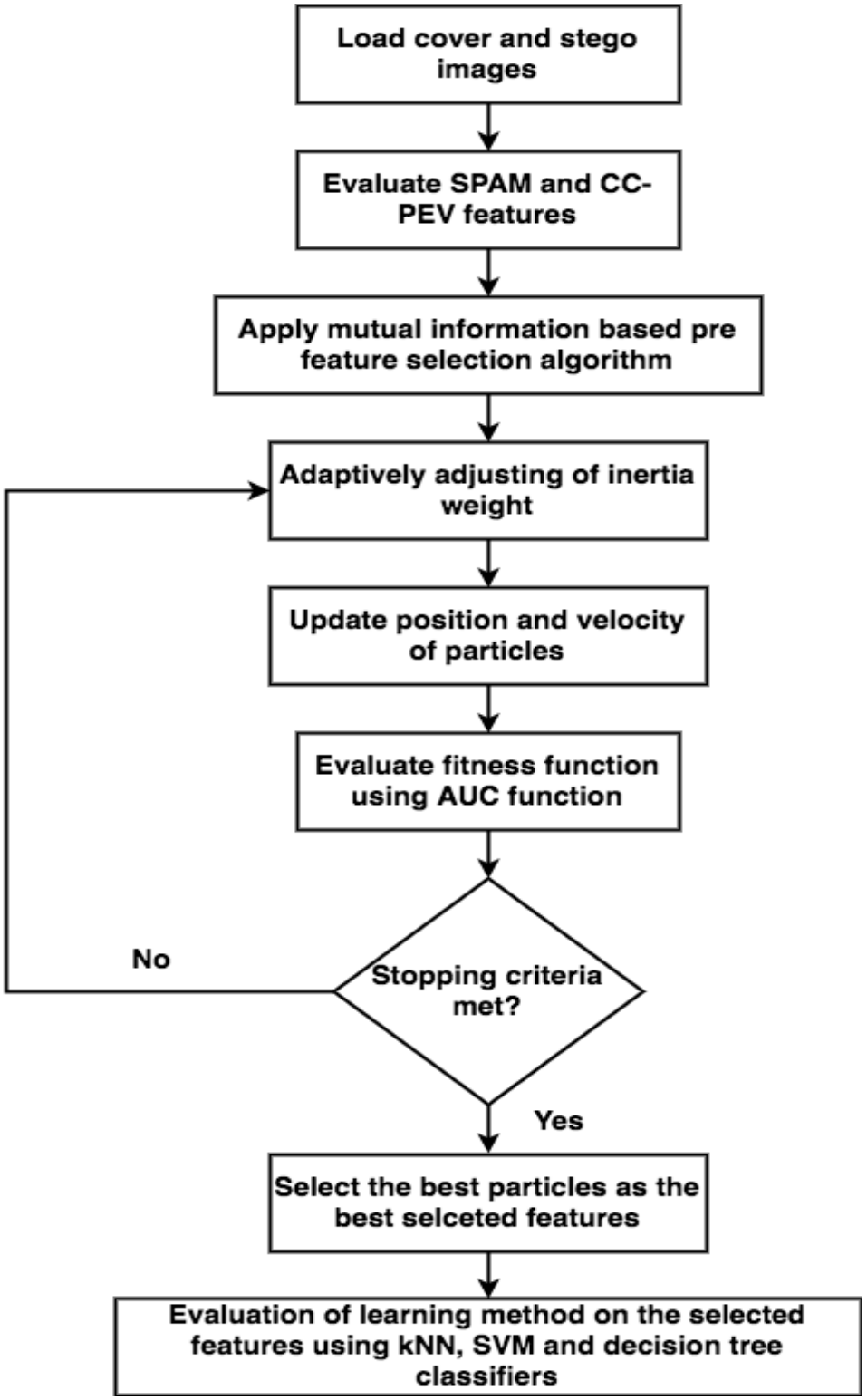

Figure 4.3: Flowchart of the proposed steganalysis classification model

Chapter 5

Experimental Results

5.1 Overview

Proposed algorithm has been applied on a dataset of 1000 cover images and corresponding Stego images taken from *BOSS* database. In this stage, experimental result and classification performance of the proposed techniques have been discussed. The experimental setup was implemented in *MATLAB* software package R2017a.Embedding rate of the hidden text in *BOSS* system is set to 0.4 per pixel. As mentioned before, two methods of feature extraction such as, *SPAM* and *CC-PEV* are applied to extract the features on the *BOSS* system for steganalysis. *SPAM* has 686 features while the length is 548 for the feature vectors of *CC-PEV*. The results evaluated of the presented method is on the basis of 10-fold cross-validation where whole dataset of both feature vectors is shuffled, out of which 10% of data is used for training process and whole dataset is tested using different classifiers. In parameter setting, the population size in *APSO* is set to 100, the parameter is 0.1, maximum iteration is 200 and both of *C1* and *C2* are set to 2.

5.2 Confusion matrix for performance evaluation

In order to check the performance of a classifier, confusion matrix is widely used which provides the true positive, true negative, false positive and false negative parameters of the tested dataset. Their correct or incorrect category classification predicted by software can be measured using sensitivity, specificity and accuracy metrics which can be derived from the outputs of confusion matrix described above. Confusion matrix gives m-by-m array that shows relationship between real and predicted categories or classes, where m is the number of classes. True Positive *(TP) i.e.* predicted positive cases that are positive in actual, False Positive *(FP)*

Figure 5.1: Cover image taken from *BOSS* database

Figure 5.2: Stego image produced after data hiding using *HUGO* algorithm based steganography

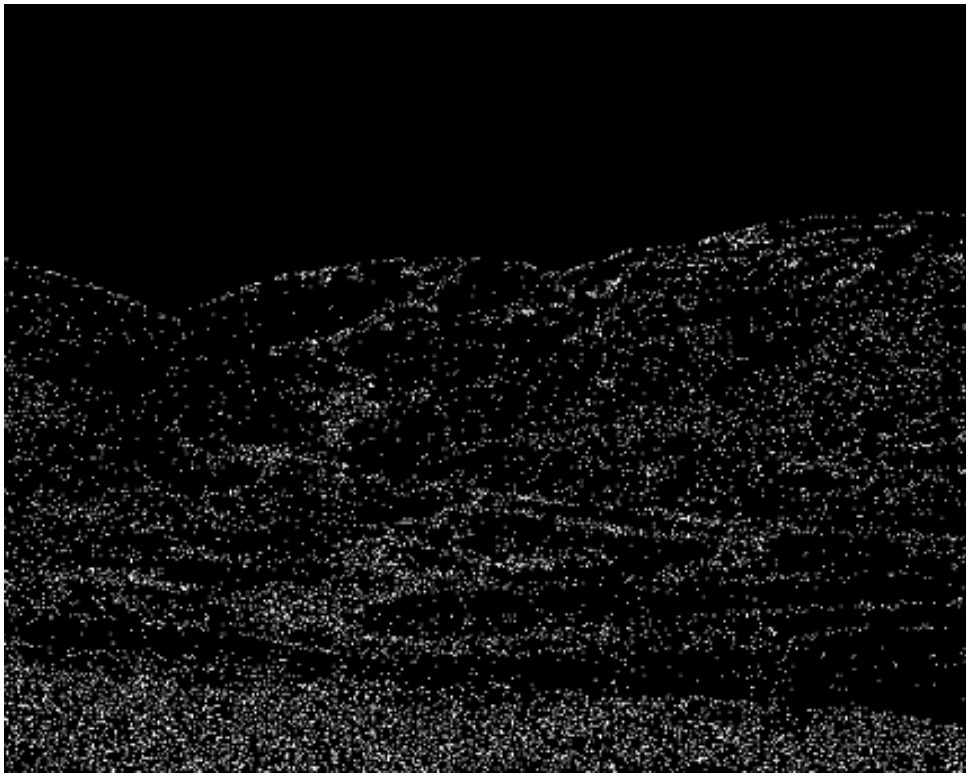

Figure 5.3: Difference image showing change in pixel values after stegnography where white pixels showing changing locations

i.e. number of predicted positive cases, True Negative *(TN) i.e.* predicted negative cases that are negative in actual and False Negative *(FN) i.e.* number of predicted negative cases that are actual in this way. Table 5.1 shows the confusion matrix for two class classifier.

Table 5.1: Confusion matrix for two class classifier

Actual Class	Predicted Class	
	Yes	No
Yes	*TP*	*FN*
No	*FP*	*TN*

5.2.1 Classification Accuracy

Classification accuracy is defined as the fraction of the number of the patterns correctly classified (TP and TN) to the total number of patterns classified.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

5.2.2 Sensitivity

The sensitivity is the fraction of the samples correctly classified as that specific species class. It is defined by equation below:

$$Sensitivity = \frac{TP}{TP + FN}$$

5.2.3 Specificity

The specificity is the fraction of normal species correctly classified as normal class. It is also called selectivity.

$$Specificity = \frac{TN}{TN + FP}$$

5.3 Results

Table 5.2 shows the values of sensitivity and specificity for stego and cover classes by using *CC-PEV+PSO* and *CC-PEV+MI+PSO* with 80 selected features using three clasifiers. Table 5.3 shows the values of sensitivity and specificity for stego and cover classes by using *CC-PEV+APSO* and *CC-PEV+MI+APSO* with 80 selected features. The classifiers used are *SVM*, *DT* and *kNN*

Table 5.2: Performance Evaluation by using *CC-PEV+PSO* and *CC-PEV+MI+PSO* with 80 selected features

Classifier used	Class	*CC-PEV* and *PSO*		*CC-PEV*, *MI* and *PSO*	
		Sensitivity	Specificity	Sensitivity	Specificity
SVM	Stego	0.6284	0.6337	0.7336	0.5851
	Cover	0.6337	0.6284	0.5851	0.7336
DT	Stego	0.669	0.6801	0.731	0.7043
	Cover	0.6801	0.669	0.7043	0.731
kNN	Stego	0.5084	0.4942	0.5041	0.501
	Cover	0.4942	0.5084	0.501	0.5041

Figure 5.4 displays the average accuracy of classification by testing features using different classifiers. Features are selected using variety of feature selection and reduction methods which is shown in the legends of the bar graphs. Number of

Table 5.3: Performance Evaluation by using *CC-PEV+APSO* and *CC-PEV+MI+APSO* with 80 selected features

Classifier used	Class	*CC-PEV* and *APSO*		*CC-PEV*, *MI* and *APSO*	
		Sensitivity	Specificity	Sensitivity	Specificity
SVM	Stego	0.6231	0.6523	0.6285	0.6695
	Cover	0.6523	0.6231	0.6695	0.6285
DT	Stego	0.6755	0.6784	0.743	0.6951
	Cover	0.6784	0.6755	0.6951	0.743
kNN	Cover	0.4994	0.5003	0.5009	0.5098
	Stego	0.5003	0.4994	0.5098	0.5009

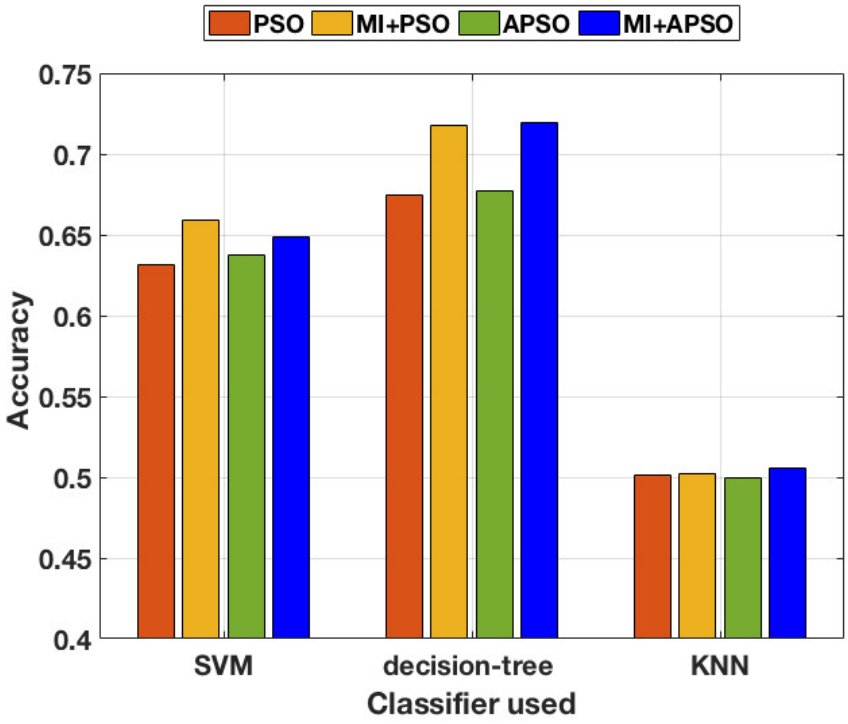

Figure 5.4: Steganalysis accuracy for *CC-PEV* features (Features selected=80)

features used for selection is 80. Bar graphs show that *MI-APSO* based feature selection gives better accuracy than other selection methods. Also Decision tree gives efficient accuracy values using all feature selection methods which is higher than other tested classifiers. Figures 5.5 and 5.6 show the sensitivity and specificity of stego images when cover images are used as secondary features using *APSO* selection method only. Figures 5.7 and 5.8 show the sensitivity and specificity of stego images when cover images are used as secondary features using *MI-APSO* selection method. Sensitivity and specificity values higher when *MI-APSO* feature reduction and selection method is used which in turn make increase in accuracy of

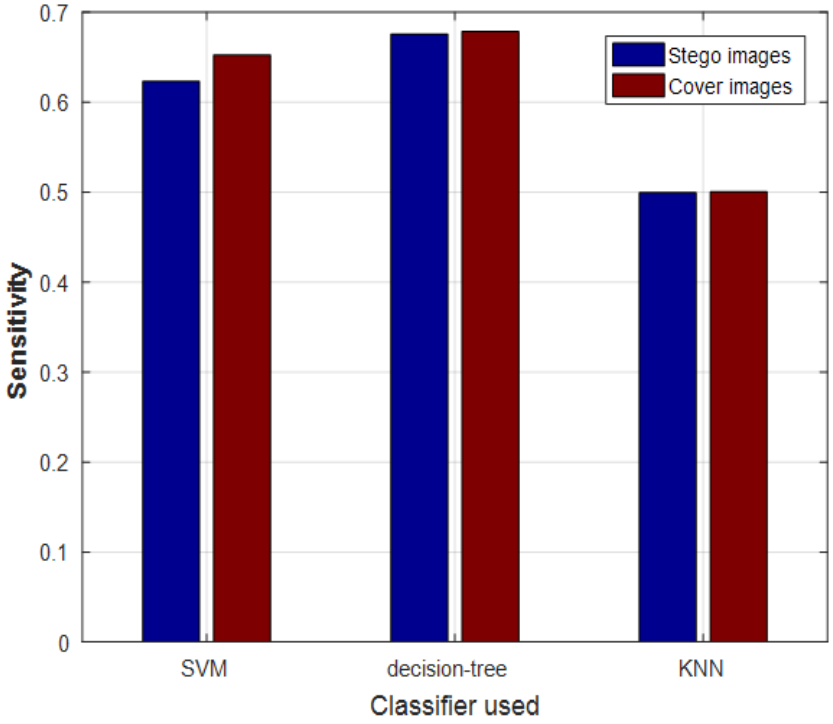

Figure 5.5: Steganalysis sensitivity for *CC-PEV* features using *APSO* (Features selected=80)

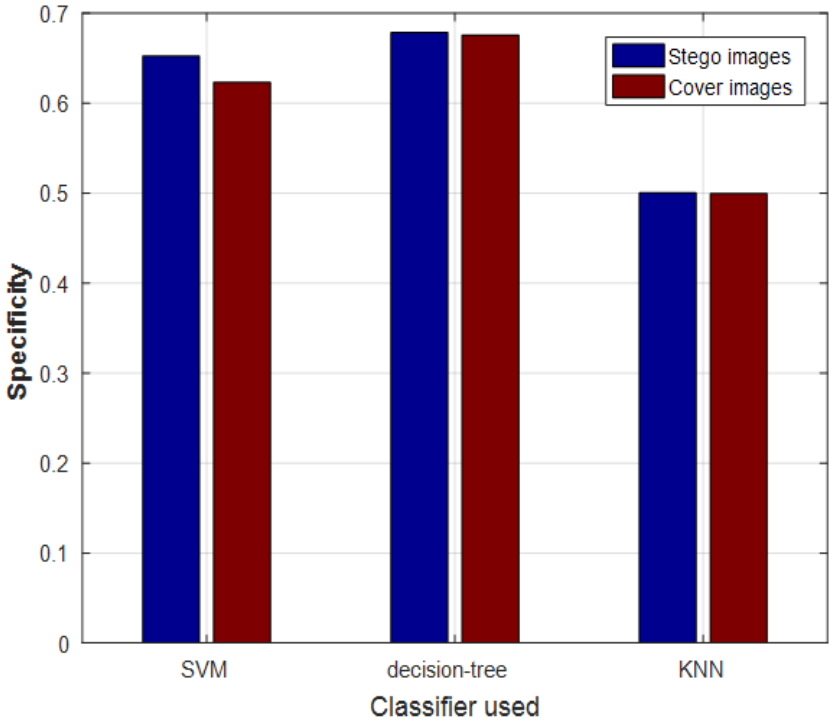

Figure 5.6: Steganalysis specificity for *CC-PEV* features using *APSO* (Features selected=80)

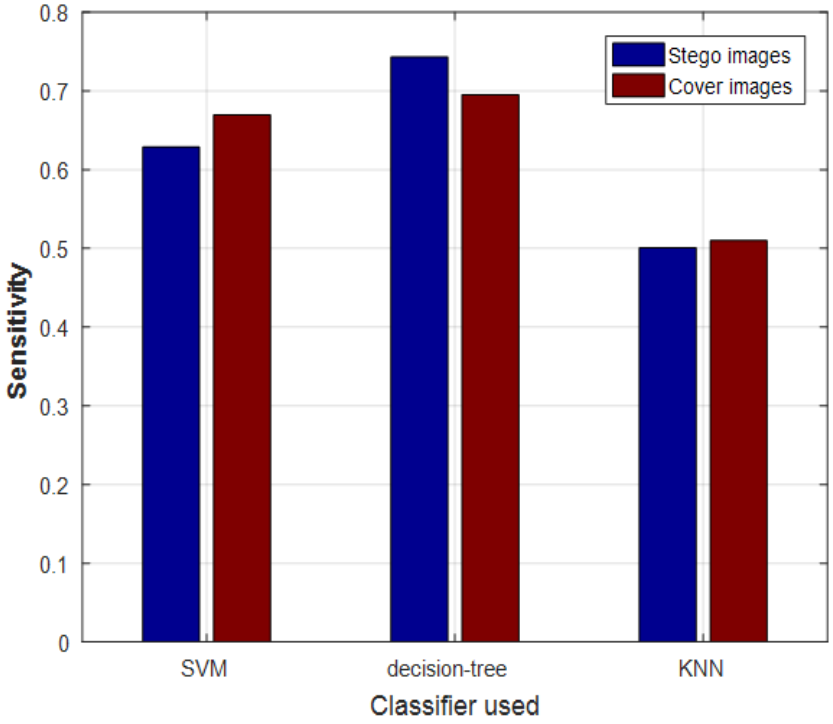

Figure 5.7: Steganalysis sensitivity for *CC-PEV* features using *MI* and *APSO* (Features selected=80)

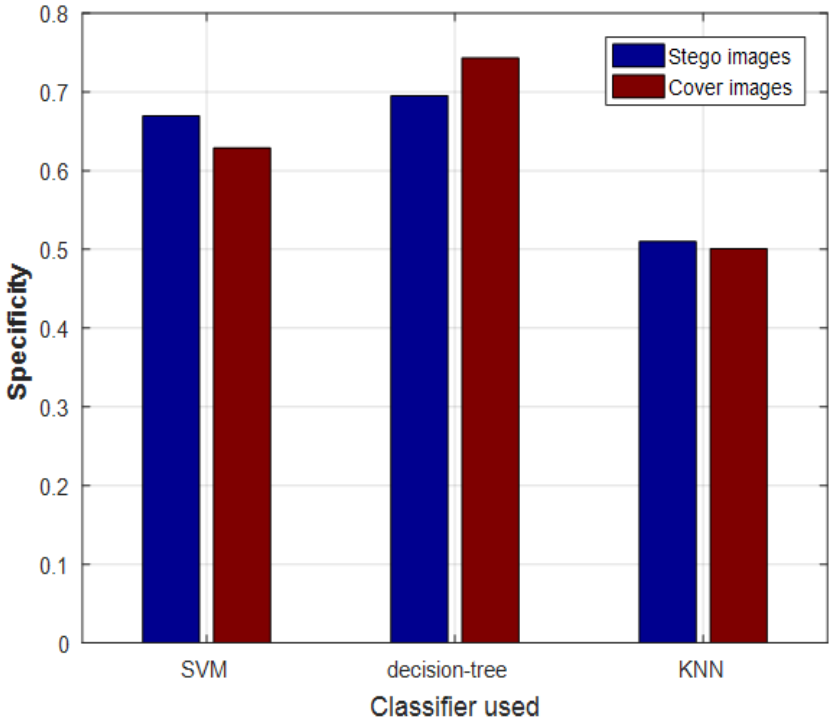

Figure 5.8: Steganalysis specificity for *CC-PEV* features using *MI* and *APSO* (Features selected=80)

the proposed feature selection method. As *MI* provides least and most seperable feature dimension to *APSO*, it chooses particle dimensions more effectively results in better convergence to the optimal solution.

Table 5.4 shows the accuracy values by using *PSO*, *MI-PSO*, *APSO* and *MI-APSO* by selecting 80 features of *CC-PEV* using three classifiers.

Table 5.5 shows the values of sensitivity and specificity for stego and cover classes by using *CC-PEV+PSO* and *CC-PEV+MI+PSO* with 100 selected features using three clasifiers. Table 5.6 shows the values of sensitivity and specificity for stego and cover classes by using *CC-PEV+APSO* and *CC-PEV+MI+APSO* with 100 selected features. Table 5.7 shows the accuracy values by using *PSO*, *MI-PSO*, *APSO* and *MI-APSO* by selecting 100 features of *CC-PEV* using three classifiers.

Table 5.4: Steganalysis Accuracy for *CC-PEV* with 80 selected feature

Classifier used	Accuracy			
	PSO	*MI-PSO*	*APSO*	*MI-APSO*
SVM	0.63105	0.65935	0.6377	0.649
DT	0.67455	0.71765	0.67695	0.71905
kNN	0.5013	0.50255	0.49985	0.50535

Table 5.5: Performance Evaluation by using *CC-PEV+PSO* and *CC-PEV+MI+PSO* with 100 selected feature

Classifier used	Class	*CC-PEV* and *PSO*		*CC-PEV*, *MI* and *PSO*	
		Sensitivity	Specificity	Sensitivity	Specificity
SVM	Stego	0.6433	0.6252	0.6285	0.6695
	Cover	0.6252	0.6433	0.6695	0.6285
DT	Stego	0.7593	0.7097	0.743	0.6951
	Cover	0.7097	0.7593	0.6951	0.743
kNN	Stego	0.4979	0.4992	0.5009	0.5098
	Cover	0.4992	0.4979	0.5098	0.5009

Table 5.8 shows the values of sensitivity and specificity for stego and cover classes by using *SPAM+PSO* and *SPAM+MI+PSO* with 80 selected features using three clasifiers. Table 5.9 shows the values of sensitivity and specificity for stego and

Table 5.6: Performance Evaluation by using *CC-PEV+APSO* and *CC-PEV+MI+APSO* with 100 selected feature

Classifier used	Class	*CC-PEV* and *APSO*		*CC-PEV*, *MI* and *APSO*	
		Sensitivity	Specificity	Sensitivity	Specificity
SVM	Stego	0.6612	0.6556	0.6456	0.7058
	Cover	0.6556	0.6612	0.7058	0.6456
DT	Stego	0.7499	0.7308	0.7478	0.7328
	Cover	0.7308	0.7499	0.7328	0.7478
kNN	Stego	0.5009	0.4982	0.4977	0.5134
	Cover	0.4982	0.5009	0.5134	0.4977

Table 5.7: Steganalysis Accuracy for *CC-PEV* with 100 selected feature

Classifier used	Accuracy			
	PSO	*MI-PSO*	*APSO*	*MI-APSO*
SVM	0.63425	0.650	0.6584	0.6757
DT	0.7345	0.718	0.74035	0.7403
kNN	0.49855	0.5060	0.49955	0.50555

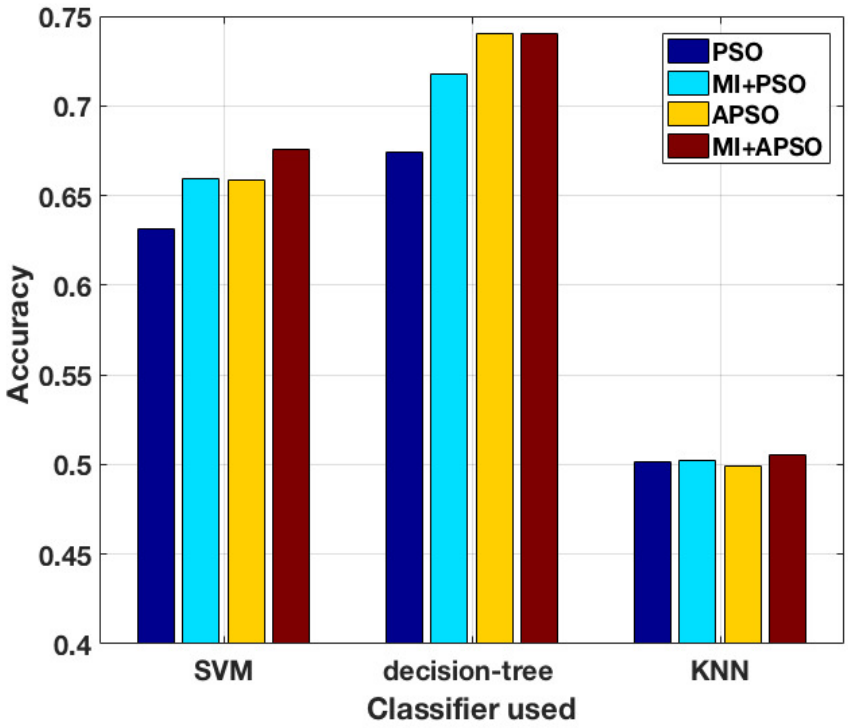

Figure 5.9: Steganalysis accuracy for *CC-PEV* features (Features selected=100)

cover classes by using *CC-PEV+APSO* and *SPAM+MI+APSO* with 80 selected features. Table 5.10 shows the accuracy values by using *PSO*, *MI-PSO*, *APSO* and *MI-APSO* by selecting 80 features of *SPAM* using three classifiers.

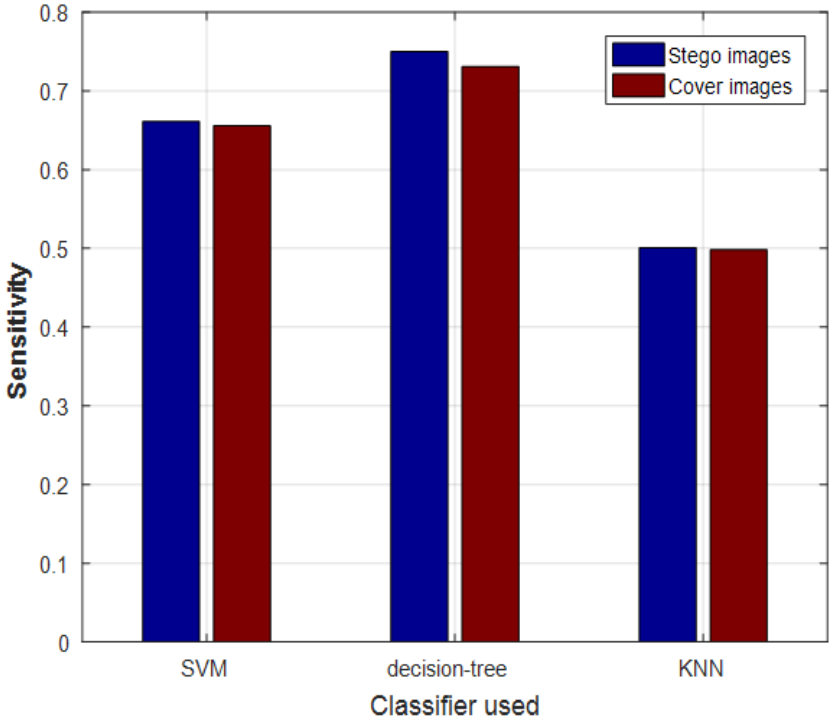

Figure 5.10: Steganalysis sensitivity for *CC-PEV* features using *APSO* (Features selected=100)

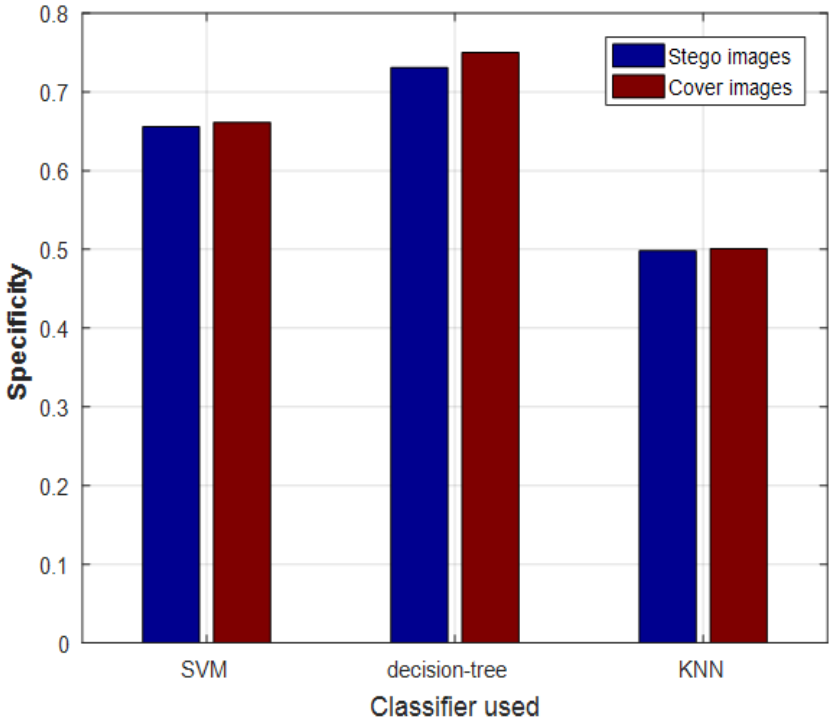

Figure 5.11: Steganalysis specificity for *CC-PEV* features using *APSO* (Features selected=100)

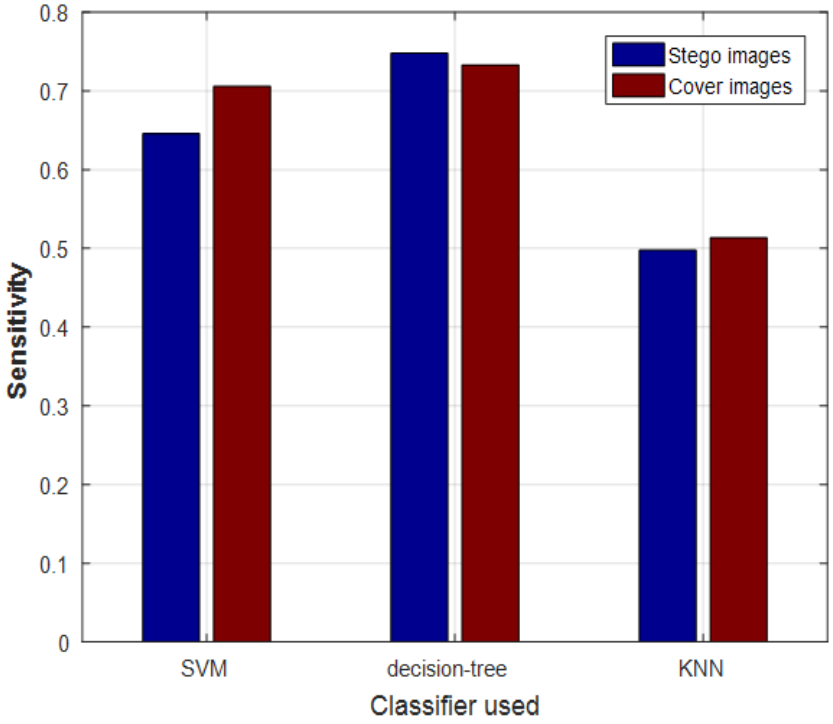

Figure 5.12: Steganalysis sensitivity for *CC-PEV* features using *MI* and *APSO* (Features selected=100)

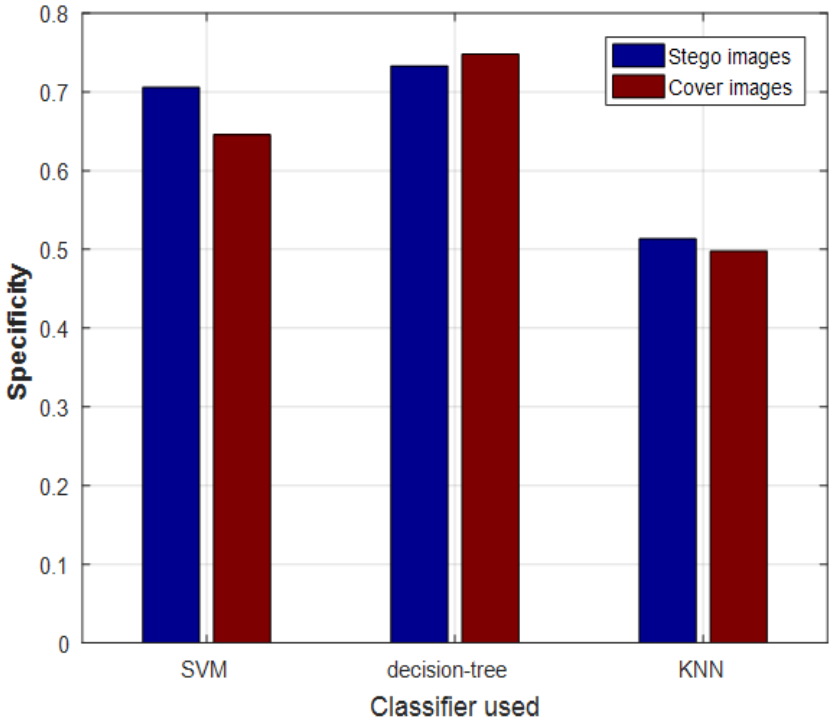

Figure 5.13: Steganalysis specificity for *CC-PEV* features using *MI* and *APSO* (Features selected=100)

Figure 5.9 displays the average accuracy of classification by testing features using different machine learning classifiers. Features are selected using variety of feature selection and reduction methods which is shown in the legends of the bar graphs. Number of features used for selection is hundred. Bar graphs show that *MI-APSO* based feature selection gives better accuracy than other selection methods. Figures 5.10 and 5.11 show the sensitivity of stego images and specificity of stego images when cover images are used as secondary features using *APSO* selection method only. Figures 5.12 and 5.13 show the sensitivity of stego images and specificity of stego images when cover images are used as secondary features using *MI-APSO* selection method. As compared to 80 numbers of selected features there is not much difference in selection of 100 features in classification accuracy. As seen the bins in decision tree classification column as the number of features increases, all algorithms decides near to best however *APSO* performs better than traditional *PSO*. Further slightest improve has been noted using *MI-PSO* than *APSO*.

Table 5.8: Performance Evaluation by using *SPAM+PSO* and *SPAM+MI+PSO* with 80 selected features

Classifier used	Class	*SPAM* and *PSO*		*SPAM*, *MI* and *PSO*	
		Sensitivity	Specificity	Sensitivity	Specificity
SVM	Stego	0.5759	0.6632	0.7448	0.642
	Cover	0.6632	0.5759	0.642	0.7448
DT	Stego	0.7206	0.7008	0.7263	0.6994
	Cover	0.7008	0.7206	0.6994	0.7263
kNN	Stego	0.5329	0.509	0.5473	0.6709
	Cover	0.509	0.5329	0.6709	0.5473

Table 5.9: Performance Evaluation by using *SPAM+APSO* and *SPAM+MI+APSO* with 80 selected features

Classifier used	Class	*SPAM* and *APSO*		*SPAM,MI* and *APSO*	
		Sensitivity	Specificity	Sensitivity	Specificity
SVM	Stego	0.5244	0.7136	0.7233	0.7004
	Cover	0.7136	0.5244	0.7004	0.7233
DT	Stego	0.7378	0.731	0.7527	0.7329
	Cover	0.731	0.7378	0.7329	0.7527
kNN	Stego	0.5186	0.5135	0.5776	0.679
	Cover	0.5135	0.5186	0.679	0.5776

Figure 5.14 displays the average accuracy of classification by testing features using

Table 5.10: Steganalysis Accuracy for *SPAM* with 80 selected feature

Classifier used	Accuracy			
	PSO	*MI-PSO*	*APSO*	*MI-APSO*
SVM	0.61955	0.6934	0.619	0.71185
DT	0.7107	0.71285	0.7344	0.7428
kNN	0.52095	0.6091	0.51605	0.6283

different machine learning classifiers when *SPAM* features are classified. 80 features are selected and reduced by four different combinations of feature reductions algorithms. Bar graphs show that *MI-APSO* based feature selection gives better accuracy than other selection methods. Also Decision tree gives efficient accuracy value using all feature selection methods which is higher than other tested classifiers. Figures 5.15 and 5.16 show the sensitivity of stego images and specificity of stego images when cover images are used as secondary features using *APSO* selection method only. Figures 5.17 and 5.18 show the sensitivity of stego images and specificity of stego images when cover images are used as secondary features using *MI+APSO* selection method. Sensitivity and specificity values higher when *MI-APSO* feature reduction and selection method is used which in turn make increase in accuracy of the proposed feature selection method. If we compare sensitivity when stego images are taken as main class, *MI-APSO* gives more sensitivity value than the *APSO* method.

Figure 5.19 displays the average accuracy of classification by testing features using different machine learning classifiers when *SPAM* features are considered. Number of features used for selection is hundred. Bar graphs show that *MI-APSO* based feature selection gives better accuracy than other selection methods. Figures 5.20 and 5.21 show the sensitivity of stego images and specificity of stego images when cover images are used as secondary features using *APSO* selection method only. Figures 5.22 and 5.23 show the sensitivity of stego images and specificity of stego images when cover images are used as secondary features using *MI+APSO* selection method. In *SVM* and *kNN* case, *MI-PSO* and *MI-APSO* shows better accuracy which is also seen in previous case when 80 features of *SPAM* method are used but decision tree better results of classification accuracy rate *APSO* and *MI-APSO* are used. As seen the bins in decision tree classification column as the

number of features increases, all algorithms decides near to best however *APSO* performs better than traditional *PSO*.

Table 5.11 shows the values of sensitivity and specificity for stego and cover classes by using *SPAM+PSO* and *SPAM+MI+PSO* with 100 selected features using three clasifiers. Table 5.12 shows the values of sensitivity and specificity for stego and cover classes by using *SPAM+APSO* and *SPAM+MI+APSO* with 100 selected features. Table 5.13 shows the accuracy values by using *PSO*, *MI-PSO*, *APSO* and *MI-APSO* by selecting 100 features of *SPAM* using three classifiers.

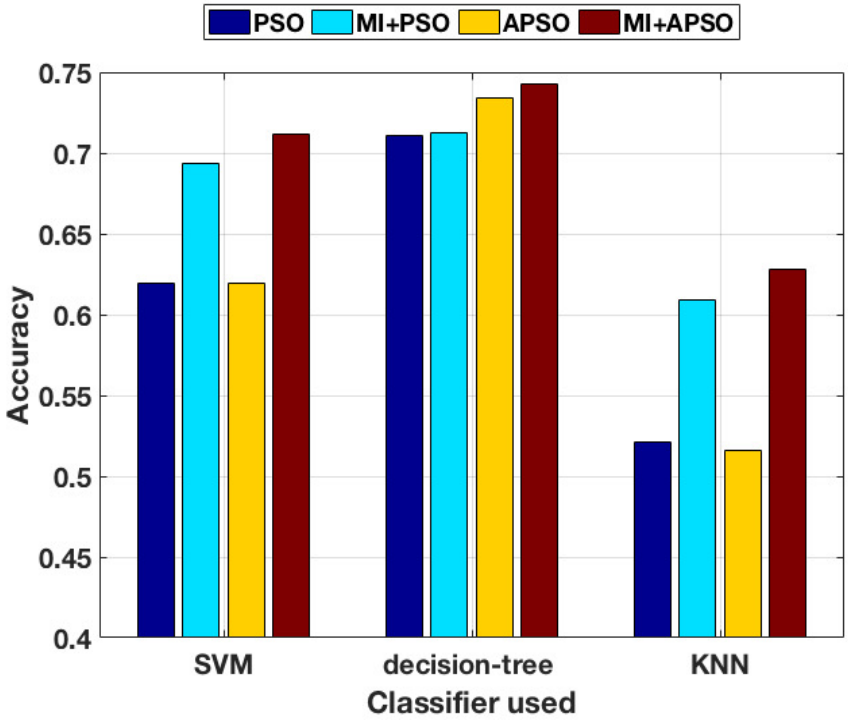

Figure 5.14: Steganalysis accuracy for *SPAM* features (Features selected=80)

Table 5.11: Performance Evaluation by using *SPAM+PSO* and *SPAM+MI+PSO* with 100 selected features

Classifier used	Class	*SPAM* and *PSO*		*SPAM,MI* and *PSO*	
		Sensitivity	Specificity	Sensitivity	Specificity
SVM	Stego	0.5244	0.7136	0.7233	0.7004
	Cover	0.7136	0.5244	0.7004	0.7233
DT	Stego	0.7378	0.731	0.7527	0.7329
	Cover	0.731	0.7378	0.7329	0.7527
kNN	Stego	0.5186	0.5135	0.5776	0.679
	Cover	0.5135	0.5186	0.679	0.5776

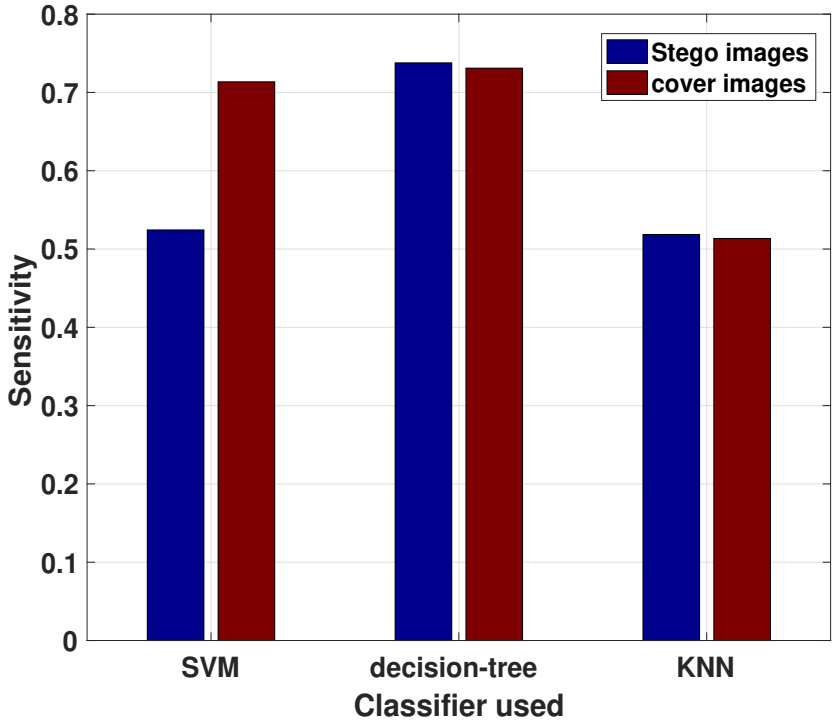

Figure 5.15: Steganalysis sensitivity for *SPAM* features using *APSO* (Features selected=80)

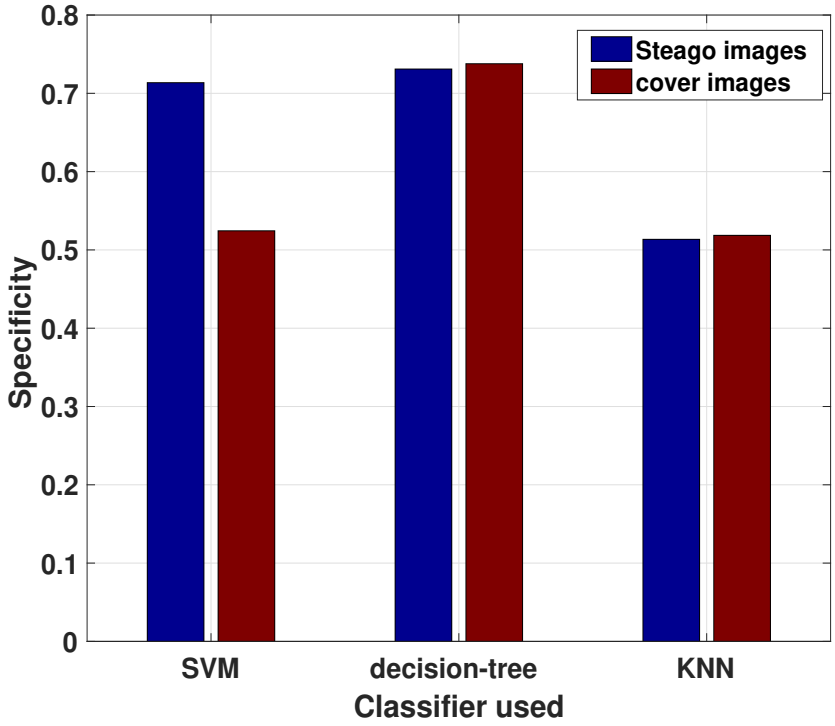

Figure 5.16: Steganalysis specificity for *SPAM* features using *APSO* (Features selected=80)

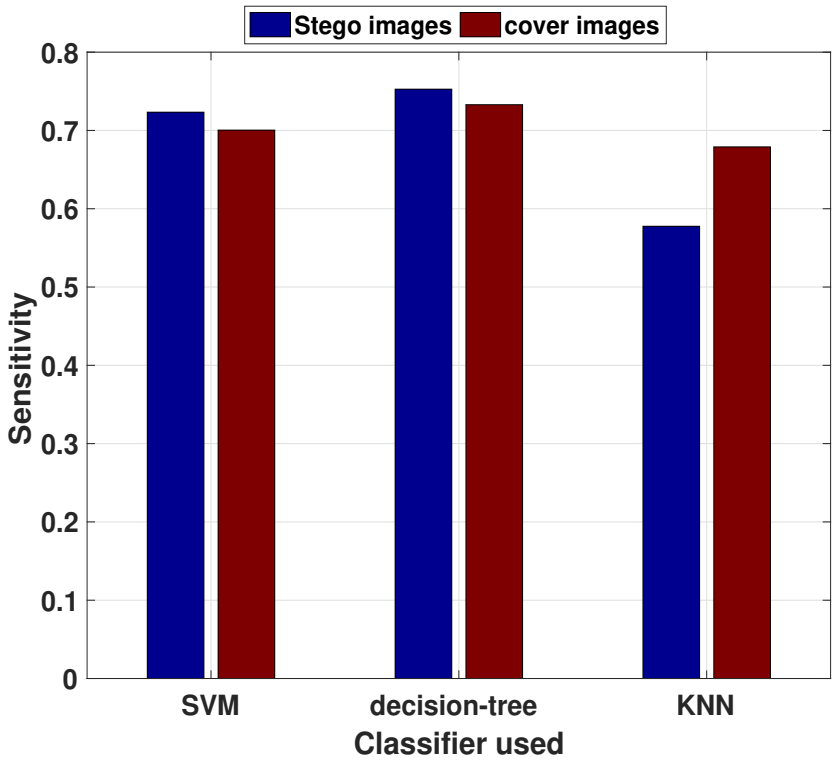

Figure 5.17: Steganalysis sensitivity for *SPAM* features using *MI* and *APSO* (Features selected=80)

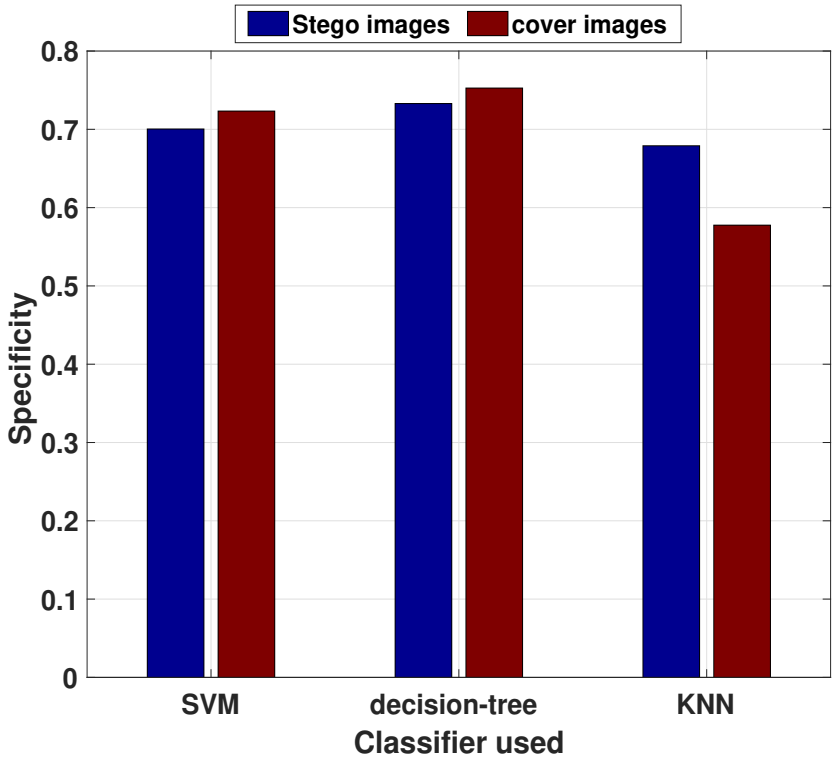

Figure 5.18: Steganalysis specificity for *SPAM* features using *MI* and *APSO* (Features selected=80)

Table 5.12: Performance Evaluation by using *SPAM+APSO* and *SPAM+MI+APSO* with 100 selected features

Classifier used	Class	*SPAM* and *APSO*		*SPAM*, *MI* and *APSO*	
		Sensitivity	Specificity	Sensitivity	Specificity
SVM	Stego	0.5791	0.6807	0.6761	0.8033
	Cover	0.6807	0.5791	0.8033	0.6761
DT	Stego	0.7456	0.7213	0.757	0.7476
	Cover	0.7213	0.7456	0.7476	0.757
kNN	Stego	0.5343	0.5188	0.5808	0.6855
	Cover	0.5188	0.5343	0.6855	0.5808

Table 5.13: Steganalysis Accuracy for *SPAM* with 100 selected features

Classifier used	Accuracy			
	PSO	*MI-PSO*	*APSO*	*MI-APSO*
SVM	0.619	0.7112	0.6299	0.7397
DT	0.7344	0.7430	0.7345	0.7523
kNN	0.51605	0.6283	0.52655	0.63315

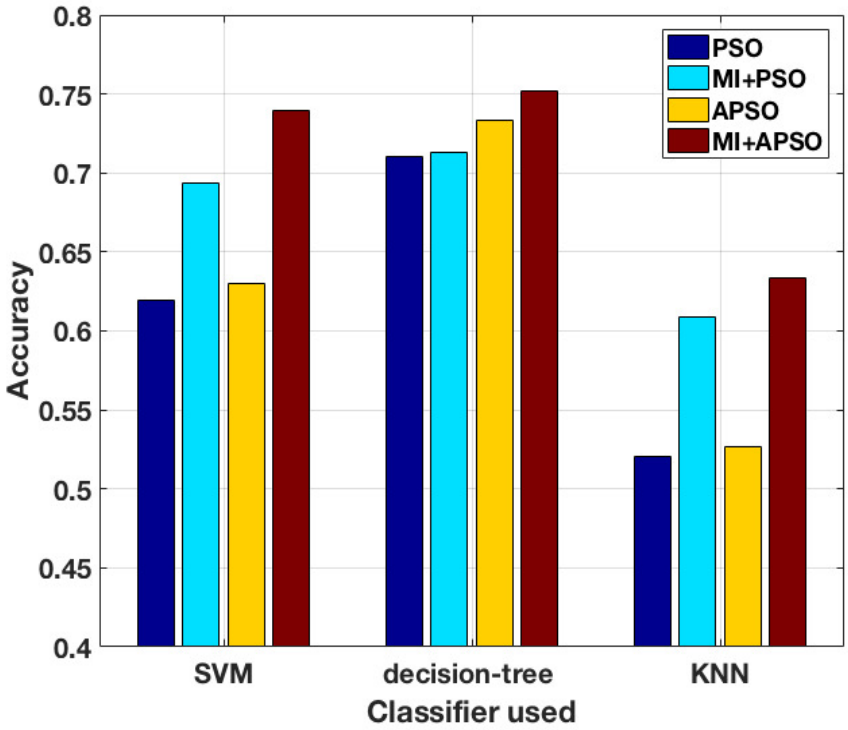

Figure 5.19: Steganalysis accuracy for *SPAM* features (Features selected=100)

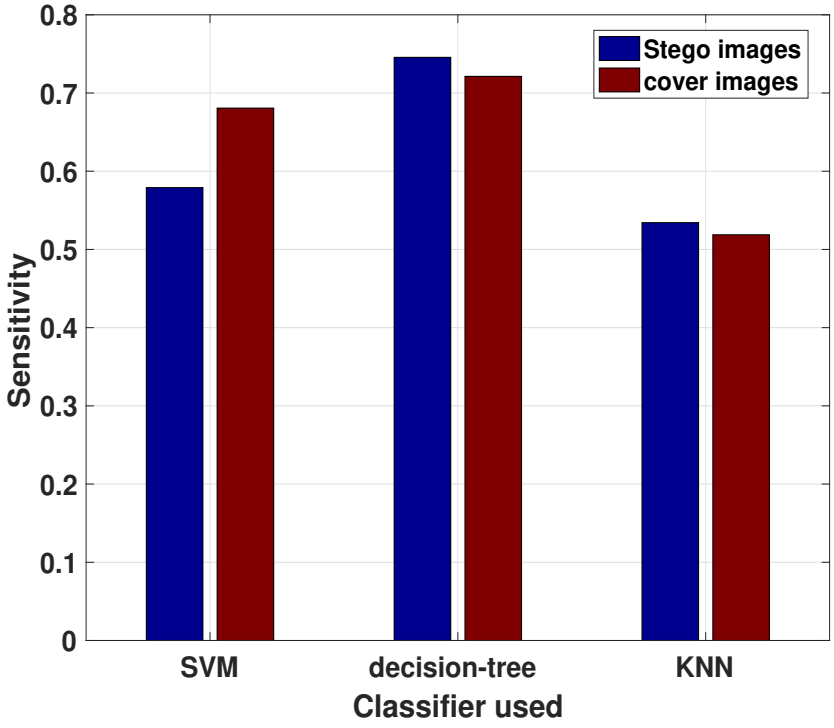

Figure 5.20: Steganalysis sensitivity for *SPAM* features using *APSO* (Features selected=100)

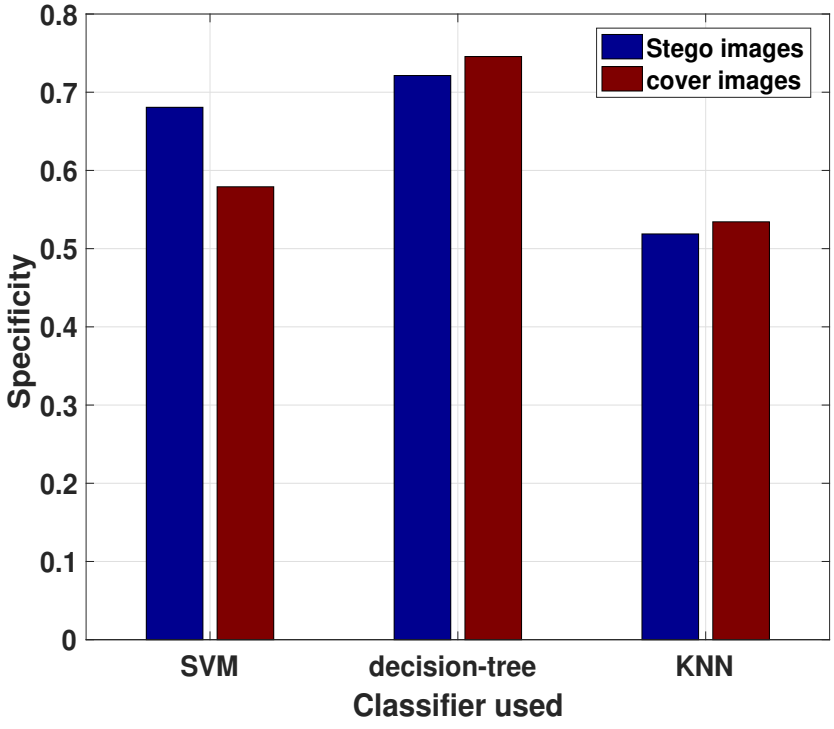

Figure 5.21: Steganalysis specificity for *SPAM* features using *APSO* (Features selected=100)

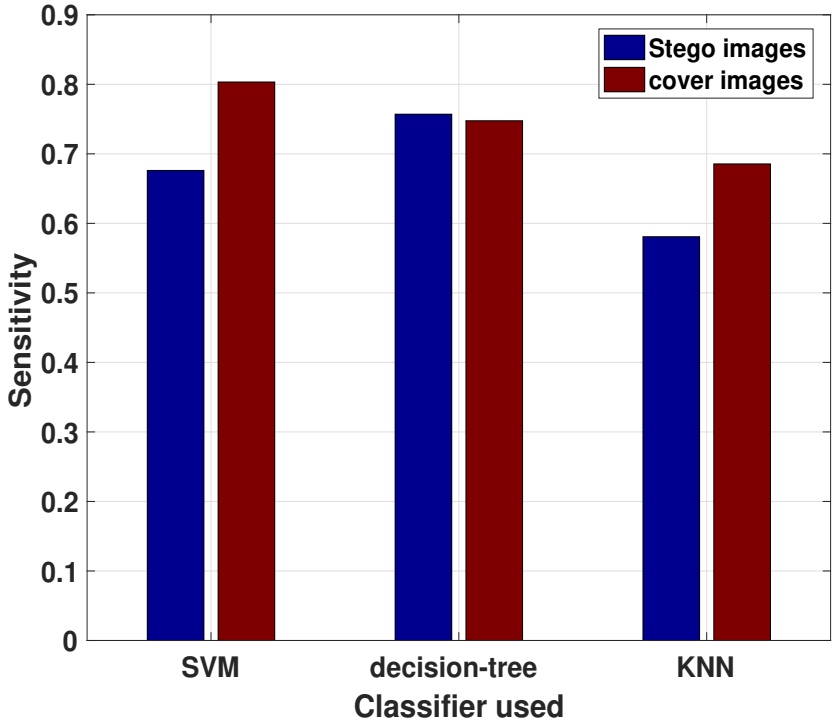

Figure 5.22: Steganalysis sensitivity for *SPAM* features using *MI* and *APSO* (Features selected=100)

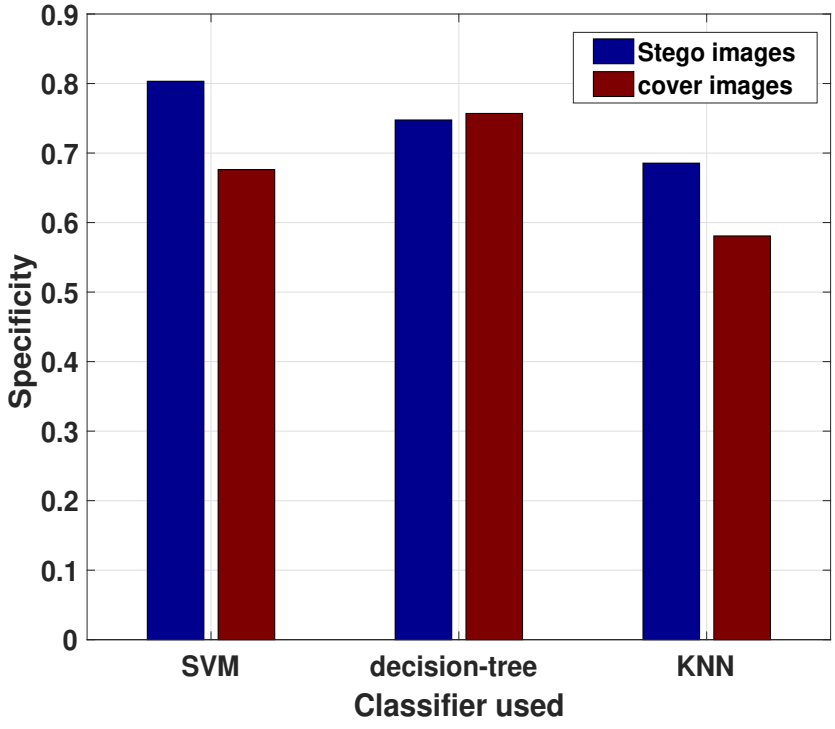

Figure 5.23: Steganalysis specificity for *SPAM* features using *MI* and *APSO* (Features selected=100)

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The goal of steganalysis is to detect the presence of secretly hidden data in an object. Digital media files contains a number of individual elements, so it is much easier to transform them in order to secretly embed data, so such files are ideal source of cover objects for steganography. Moreover, except few cases, detection can not be made only on the basis of estimates of the underlying probability distributions of statistics extracted from cover and stego objects. Instead, detection is usually cast as a supervised classification problem implemented using machine learning. For steganalysis, many feature extraction algorithms are used in which *SPAM* and *CC-PEV* found effective in the literature. These feature extraction algorithms have high feature dimension space in which some dimensions have similar values for the classes which reduce the effectiveness of the classifiers and low accuracy is achieved by the classification step of steganalysis. Also computation time and complexity of the system increases with the larger number of fed feature dimension. In order to overcome these drawbacks, feature selection phase need to be introduced in between feature extraction and classification phase. In this work, two feature selection methods has been introduced in which *MI* algorithm first sorts the feature dimensions according to mutual information value. It is a filter-based feature selection approach that analyzes the *MI* between discrete features and class targets. Selected features from this phase are further fed to adaptive *PSO* to further decrease the feature space which used *AUC* value of a binary classifier to differentiate features of stego and cover images. Selected feature dimensions proposed by *APSO* are then further validated and tested using different machine learning algorithms named as *DT*, *k-NN* and *SVM*. Experimental results shows high classification accuracy when proposed *MI-APSO* feature sections are used. Results of the proposed system is compared with the traditional *PSO* for classification. Further among the three used machine learning classifiers, *DT* gives more

accuracy in classification which is approximate 74% when 100 number of *SPAM* and *CC-PEV* features are selected by *APSO*.

6.2 Future Scope

In this work, only *SPAM* and *CC-PEV* features are explored. Other feature extraction methods can be explored along with these. There are number of optimization algorithms i.e. gravitational search algorithm, cuckoo search, Ant Colony Optimization *(ACO)*, *etc*. Behavior and effectiveness of these optimization algorithms can be compared with existed methods.

References

[1] R. J. Anderson and F. A. Petitcolas, "On the limits of steganography," *IEEE Journal on selected areas in communications*, vol. 16, no. 4, pp. 474–481, 1998.

[2] C. Cachin, "An information-theoretic model for steganography," in *International Workshop on Information Hiding*. Springer, 1998, pp. 306–318.

[3] C.-c. Chang and C.-j. Lin, "Libsvm: A library for support vector machines. software available at http://www. csie. ntu. edu. tw/cjlin/libsvm," 2001.

[4] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *Journal of machine learning research*, vol. 9, no. Aug, pp. 1871–1874, 2008.

[5] C. Chen and Y. Q. Shi, "Jpeg image steganalysis utilizing both intrablock and interblock correlations," in *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*. IEEE, 2008, pp. 3029–3032.

[6] J. Kodovskỳ and J. Fridrich, "Calibration revisited," in *Proceedings of the 11th ACM workshop on Multimedia and security*. ACM, 2009, pp. 63–74.

[7] T. Pevny, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *IEEE Transactions on information Forensics and Security*, vol. 5, no. 2, pp. 215–224, 2010.

[8] J. Kodovskỳ and J. Fridrich, "Steganalysis in high dimensions: Fusing classifiers built on random subspaces," in *Media Watermarking, Security, and Forensics III*, vol. 7880. International Society for Optics and Photonics, 2011, p. 78800L.

[9] J. Kodovskỳ, J. J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media." *IEEE Trans. Information Forensics and Security*, vol. 7, no. 2, pp. 432–444, 2012.

[10] Q. Liu, "Steganalysis of dct-embedding based adaptive steganography and yass," in *Proceedings of the thirteenth ACM multimedia workshop on Multimedia and security*. ACM, 2011, pp. 77–86.

[11] S. Cho, B.-H. Cha, M. Gawecki, and C.-C. J. Kuo, "Block-based image steganalysis: Algorithm and performance evaluation," *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 846–856, 2013.

[12] X. Hou, T. Zhang, G. Xiong, Z. Lu, and K. Xie, "A novel steganalysis framework of heterogeneous images based on gmm clustering," *Signal Processing: Image Communication*, vol. 29, no. 3, pp. 385–399, 2014.

[13] P. Pathak and S. Selvakumar, "Blind image steganalysis of jpeg images using feature extraction through the process of dilation," *Digital Investigation*, vol. 11, no. 1, pp. 67–77, 2014.

[14] S. Akhavan, M. A. Akhaee, and S. Sarreshtedari, "Images steganalysis using garch model for feature selection," *Signal Processing: Image Communication*, vol. 39, pp. 75–83, 2015.

[15] X. Kong, C. Feng, M. Li, and Y. Guo, "Iterative multi-order feature alignment for jpeg mismatched steganalysis," *Neurocomputing*, vol. 214, pp. 458–470, 2016.

[16] H. Sajedi, "Steganalysis based on steganography pattern discovery," *Journal of Information Security and Applications*, vol. 30, pp. 3–14, 2016.

[17] Z. Li and A. G. Bors, "Steganalysis of 3d objects using statistics of local feature sets," *Information Sciences*, vol. 415, pp. 85–99, 2017.

[18] B. Feng, J. Weng, W. Lu, and B. Pei, "Steganalysis of content-adaptive binary image data hiding," *Journal of Visual Communication and Image Representation*, vol. 46, pp. 119–127, 2017.

[19] R. Ramesh, C. Gomathy, D. Vaishali *et al.*, "Bio inspired optimization for universal spatial image steganalysis," *Journal of Computational Science*, vol. 21, pp. 182–188, 2017.

[20] K. Karampidis, E. Kavallieratou, and G. Papadourakis, "A review of image

steganalysis techniques for digital forensics," *Journal of information security and applications*, vol. 40, pp. 217–235, 2018.

[21] S. Geetha and N. Kamaraj, "Optimized image steganalysis through feature selection using mbega," *arXiv preprint arXiv:1008.2824*, 2010.

[22] M. J. Zomorodian, A. Adeli, M. Sinaee, and S. Hashemi, "Improving nearest neighbor classification by elimination of noisy irrelevant features," in *Asian Conference on Intelligent Information and Database Systems*. Springer, 2012, pp. 11–21.

[23] G. Chen, Q. Chen, D. Zhang, and W. Zhu, "Particle swarm optimization feature selection for image steganalysis," in *Digital Home (ICDH), 2012 Fourth International Conference on*. IEEE, 2012, pp. 304–308.

[24] A. Adeli, A. Ghorbani-Rad, M. J. Zomorodian, M. Neshat, and S. Mozaffari, "Improving nearest neighbor classification using particle swarm optimization with novel fitness function," in *International Conference on Computational Collective Intelligence*. Springer, 2012, pp. 365–372.

[25] D. Yazdani, B. Nasiri, A. Sepas-Moghaddam, and M. R. Meybodi, "A novel multi-swarm algorithm for optimization in dynamic environments based on particle swarm optimization," *Applied Soft Computing*, vol. 13, no. 4, pp. 2144–2158, 2013.

[26] F. G. Mohammadi and M. S. Abadeh, "Image steganalysis using a bee colony based feature selection algorithm," *Engineering Applications of Artificial Intelligence*, vol. 31, pp. 35–43, 2014.

[27] J.-c. Lu, F.-l. Liu, and X.-y. Luo, "Selection of image features for steganalysis based on the fisher criterion," *Digital Investigation*, vol. 11, no. 1, pp. 57–66, 2014.

[28] A. Wu, G. Feng, X. Zhang, and Y. Ren, "Unbalanced jpeg image steganalysis via multiview data match," *Journal of visual communication and image representation*, vol. 34, pp. 103–107, 2016.

[29] D. Lerch-Hostalot and D. Megías, "Unsupervised steganalysis based on artifi-

cial training sets," *Engineering Applications of Artificial Intelligence*, vol. 50, pp. 45–59, 2016.

[30] V. Rostami and A. S. Khiavi, "Particle swarm optimization based feature selection with novel fitness function for image steganalysis," in *Artificial Intelligence and Robotics (IRANOPEN), 2016*. IEEE, 2016, pp. 109–114.

[31] X. Hou, T. Zhang, L. Ji, and Y. Wu, "Combating highly imbalanced steganalysis with small training samples using feature selection," *Journal of Visual Communication and Image Representation*, vol. 49, pp. 243–256, 2017.

[32] F. G. Mohammadi and H. Sajedi, "Region based image steganalysis using artificial bee colony," *Journal of Visual Communication and Image Representation*, vol. 44, pp. 214–226, 2017.

[33] S. Veena and S. Arivazhagan, "Quantitative steganalysis of spatial lsb based stego images using reduced instances and features," *Pattern Recognition Letters*, vol. 105, pp. 39–49, 2018.

[34] A. Adeli and A. Broumandnia, "Image steganalysis using improved particle swarm optimization based feature selection," *Applied Intelligence*, pp. 1–14, 2018.

[35] F. G. Mohammadi and M. S. Abadeh, "A new metaheuristic feature subset selection approach for image steganalysis," *Journal of Intelligent & Fuzzy Systems*, vol. 27, no. 3, pp. 1445–1455, 2014.

[36] H. Sajedi, "Image steganalysis using artificial bee colony algorithm," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 29, no. 5, pp. 949–966, 2017.

[37] T. Filler, T. Pevny, S. Craver, and A. Ker, *Information Hiding: 13th International Conference, IH 2011, Prague, Czech Republic, May 18-20, 2011, Revised Selected Papers*. Springer Science & Business Media, 2011, vol. 6958.

[38] R. Kennedy, "J. and eberhart, particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks IV, pages*, vol. 1000, 1995.

[39] J. Riget and J. S. Vesterstrøm, "A diversity-guided particle swarm optimizer-the arpso," *Dept. Comput. Sci., Univ. of Aarhus, Aarhus, Denmark, Tech. Rep*, vol. 2, p. 2002, 2002.

List of Publications

1. Jasmanpreet Kaur, Singara Singh "*Feature Selection using Mutual Information and Adaptive Particle Swarm Optimization for Image Steganalysis*", IEEE 7th International Conference on Realiability, Infocom Technologies and Optimization [Accepted].